NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

GEOLOCATION OF LTE SUBSCRIBER STATIONS BASED ON THE TIMING ADVANCE RANGING PARAMETER

by

Leslie A. Jarvis, Jr.

December 2010

Thesis Co-Advisors:	John C. McEachen Herschel H. Loomis

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2010	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE Geolocation of LTE Subscriber Stations Based on the Timing Advance Ranging Parameter	**5. FUNDING NUMBERS**
6. AUTHOR(S) Leslie A. Jarvis, Jr.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number _____N.A.___________.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

The possibility of geolocating a Long Term Evolution (LTE) subscriber station based on the timing advance ranging parameter within the network signal internals is investigated in this thesis. The basic approach to geolocation based on radial distances from multiple base stations is outlined. Specifics of the timing parameters used during LTE network entry are examined as they relate to calculating these distances. Computer simulation is used to demonstrate expected geolocation accuracy in multiple base station networks when estimating likely locations of subscriber stations on a two-dimensional coordinate mapping system. Computer simulation is further refined to demonstrate expected geolocation accuracy in multiple base station networks when estimating likely locations of subscriber stations on a three-dimensional coordinate mapping scheme. The possibility of fixes with ten times greater accuracy than in previous results in literature are shown by applying timing advance techniques to Global System for Mobile communications networks when using a two-dimensional coordinate mapping scheme. Accuracy capable of being within 50 centimeters when using a three-dimensional coordinate mapping scheme, comparable to the accuracy in Global Positioning System technologies, are also shown.

14. SUBJECT TERMS 3GPP, Long Term Evolution, LTE, Geolocation, Ranging, Timing Advance	**15. NUMBER OF PAGES** 123
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

GEOLOCATION OF LTE SUBSCRIBER STATIONS BASED ON THE TIMING ADVANCE RANGING PARAMETER

Leslie A. Jarvis, Jr.
Lieutenant, United States Navy
B.S., The Citadel, 2006

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2010

Author: Leslie A. Jarvis, Jr.

Approved by: John C. McEachen
Thesis Co-Advisor

Herschel H. Loomis, Jr.
Thesis Co-Advisor

Ralph C. Robertson
Chairman, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The possibility of geolocating a Long Term Evolution (LTE) subscriber station based on the timing advance ranging parameter within the network signal internals is investigated in this thesis. The basic approach to geolocation based on radial distances from multiple base stations is outlined. Specifics of the timing parameters used during LTE network entry are examined as they relate to calculating these distances. Computer simulation is used to demonstrate expected geolocation accuracy in multiple base station networks when estimating likely locations of subscriber stations on a two-dimensional coordinate mapping system. Computer simulation is further refined to demonstrate expected geolocation accuracy in multiple base station networks when estimating likely locations of subscriber stations on a three-dimensional coordinate mapping scheme. The possibility of fixes with ten times greater accuracy than in previous results in literature are shown by applying timing advance techniques to Global System for Mobile communications networks when using a two-dimensional coordinate mapping scheme. Accuracy capable of being within 50 centimeters when using a three-dimensional coordinate mapping scheme, comparable to the accuracy in Global Positioning System technologies, also is shown.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION....................1
A. BACKGROUND: LTE AND WHY WE CARE....................1
B. OBJECTIVE: GEOLOCATION....................3
C. RELATED WORK: METHODS OF GEOLOCATION....................4
D. APPROACH....................6
E. THESIS ORGANIZATION....................7

II. INVESTIGATION OF LTE WORKINGS....................9
A. UPLINK AND DOWNLINK....................9
B. RADIO FRAME STRUCTURES AND TIMING....................13
C. MEDIUM ACCESS CONTROL....................14
D. TIMING ADVANCE CALCULATIONS....................17

III. TWO-DIMENSIONAL SIMULATIONS....................19
A. METHOD OF APPROXIMATIONS....................19
B. LIKELY LOCATION CALCULATIONS....................21
C. TWO BASE STATION SIMULATION....................25
D. MULTIPLE BASE STATION SIMULATION....................27

IV. THREE-DIMENSIONAL SIMULATIONS....................35
A. METHOD OF APPROXIMATIONS....................35
B. LIKELY LOCATION CALCULATIONS....................36
C. THREE BASE STATION SIMULATION....................39
D. FOUR BASE STATION SIMULATIONS....................42
E. MULTIPLE BASE STATION SIMULATIONS....................47

V. CONCLUSIONS AND RECOMMENDATIONS....................57
A. CONCLUSIONS....................57
B. RECOMMENDATIONS....................58

APPENDIX A. LTE PARAMETERS FOR DOWNLINK TRANSMISSION SCHEME....................59

APPENDIX B. TWO-DIMENSIONAL SIMULATIONS....................61
A. TWO BASE STATIONS THROUGH VARYING ANGLES....................61
1. Two Base Stations Through Varying Angles MATLAB Code....................61
2. Two Base Stations Through Varying Angles Example Plots....................64
B. MULTIPLE BASE STATIONS....................65
1. Multiple Base Stations MATLAB Code....................65
2. Random Angle and Distance Example Plots....................71
3. Evenly Spaced Angles with Random Distance Example Plots....................72
4. Evenly Spaced Angles with Fixed Distance Example Plots....................73
5. Evenly Spaced Angles with Fixed Distance Example Plots, Standard Deviation Applied....................74

APPENDIX C. THREE-DIMENSIONAL SIMULATIONS....................75

A. THREE BASE STATIONS 75

1. Three Base Stations Through Varying Angle MATLAB Code 75
2. Three Base Stations Through Varying Angles Example Plots 79

B. FOUR BASE STATIONS 80

1. Four Base Stations MATLAB Code 80
2. Random Angles, Distances and Heights Example Plots 86
3. Even Angles with Random Distances and Heights Example Plots 87
4. Even Angles with Fixed Distances and Random Heights Example Plots 88
5. Even Angles with Fixed Distances and Fixed Heights Example Plots, Standard Deviation Applied 89

C. MULTIPLE BASE STATIONS 90

1. Multiple Base Stations MATLAB Code 90
2. Random Angles, Distances and Heights Example Plots 97
3. Even Angles with Random Distances and Heights Example Plots 98
4. Even Angles with Fixed Distances and Random Heights Example Plots 99
5. Even Angles with Fixed Distances and Fixed Heights Example Plots 100

LIST OF REFERENCES 101

INITIAL DISTRIBUTION LIST 103

LIST OF FIGURES

Figure 1. LTE Forum LTE Deployment map (From [5])..3

Figure 2. Illustration of overlapping timing advance range rings (From [8]). 6

Figure 3. Functional commonality between SC-FDMA and OFDMA signal chains (From [14])..11

Figure 4. Principles of TDD and FDD modes of operation (From [14]). 12

Figure 5. Uplink timing control (From [14])..13

Figure 6. Frame structure type 1 (From [15]). 14

Figure 7. Uplink-downlink timing relation (From [15]). 14

Figure 8. Example of MAC PDU consisting of MAC header, MAC control elements, MAC SDUs and padding (From [16]). 15

Figure 9. Example of MAC PDU consisting of MAC RARs (From [16]). 16

Figure 10. MAC RAR (From [16]). 16

Figure 11. Timing Advance Command MAC control element (From [16]). 17

Figure 12. Uniform distribution of Δr 21

Figure 13. Decision process for geolocation method of intersecting radii.......................22

Figure 14. Use of triangles to calculate intersection of circles. 23

Figure 15. Approximation error analysis of two eNodeBs varying angle simulation.......26

Figure 16. Approximation error analysis of two eNodeBs varying angle simulation, standard deviation applied. 27

Figure 17. Inaccurate approximation situation with a 3 eNodeB network.......................29

Figure 18. Average Distance from Estimate to UE with Multiple eNodeBs. 31

Figure 19. Error analysis of multiple eNodeBs with even angles and fixed radial distances...31

Figure 20. Average distance from estimate to UE with multiple eNodeBs, standard deviation applied...32

Figure 21. Circular Error Probable from multiple eNodeB simulations. 33

Figure 22. Circular Error Probable from multiple eNodeB simulations, standard deviation applied...34

Figure 23. Illustration of difference between 2D and 3D ranges (After [18]). 35

Figure 24. Three-dimensional average distance from estimate to UE with 3 eNodeBs. ..41

Figure 25. Mean Radial Spherical Error from 3 eNodeBs simulation. 41

Figure 26. Example of gross three-dimensional approximation error with a 4 eNodeB network. 43

Figure 27. Three-dimensional estimate error analysis with 4 eNodeBs...........................44

Figure 28. Three-dimensional MRSE error analysis with 4 eNodeBs. 44

Figure 29. Three-dimensional estimate error analysis with 4 eNodeBs, standard deviation applied...45

Figure 30. Three-dimensional MRSE error analysis with 4 eNodeBs, standard deviation applied...46

Figure 31. Example of an accurate location estimate with a 4 eNodeB network using approximation X-Y coordinates and disregarding the gross Z-coordinate value...47

Figure 32. Three-dimensional estimate error analysis with multiple eNodeBs. 49
Figure 33. Three-dimensional MRSE error analysis with multiple eNodeBs..................49
Figure 34. Three-dimensional MRSE error analysis with multiple eNodeBs (enhanced view). 50
Figure 35. Three-dimensional estimate error analysis with multiple eNodeBs, standard deviation applied. 51
Figure 36. Three-dimensional MRSE error analysis with multiple eNodeBs, standard deviation applied. 52
Figure 37. Three-dimensional MRSE error analysis with multiple eNodeBs, standard deviation applied (enhanced view). 52
Figure 38. Three-dimensional estimate error analysis with multiple eNodeBs, standard deviation applied and disregarding approximation Z-coordinate in the four eNodeB network..53
Figure 39. Three-dimensional MRSE error analysis with multiple eNodeBs, standard deviation applied and disregarding approximation Z-coordinate in the four eNodeB network. 54
Figure 40. Three-dimensional MRSE error analysis with multiple eNodeBs, standard deviation applied and disregarding approximation Z-coordinate in the four eNodeB network (enhanced view)..55
Figure 41. Sample plots from 2 eNodeB simulation with varying angles and distances. 64
Figure 42. Sample multiple eNodeB plots with random angles and distances. 71
Figure 43. Sample multiple eNodeB plots with even angles and random distances.........72
Figure 44. Sample multiple eNodeB plots with even angles and fixed 1 km distance. 73
Figure 45. Sample multiple eNodeB plots with even angles and fixed 1 km distance, standard deviation applied. 74
Figure 46. Sample plots from three base station simulation with varying angles, random distances and random eNodeB heights. 79
Figure 47. Sample 4 eNodeB plots with random angles, distances and heights. 86
Figure 48. Sample 4 eNodeB plots with evenly spaced angles, random distances and heights. 87
Figure 49. Sample 4 eNodeB plots with evenly spaced angles, fixed 1732m distances and random heights. 88
Figure 50. Sample 4 eNodeB plots with evenly spaced angles, fixed 1732m distances and fixed 305m eNodeB heights, standard deviation applied.........................89
Figure 51. Sample multiple eNodeB plots with random angles, distances and heights....97
Figure 52. Sample multiple eNodeB plots with evenly spaced angles, random distances and heights..98
Figure 53. Sample multiple eNodeB plots with evenly spaced angles, fixed 1732m distances and random heights. 99
Figure 54. Sample multiple eNodeB plots with evenly spaced angles, fixed 1732m distances and fixed 305m eNodeB heights. 100

LIST OF ACRONYMS AND ABBREVIATIONS

2G	Second Generation
3G	Third Generation
3GPP	Third Generation Partnership Project
4G	Fourth Generation
AOA	Angle of Arrival
BS	Base Station
CDMA	Code Division Multiple Access
CEP	Circular Error Probable
CP	Cyclic Prefix
C-RNTI	Cell Radio Network Temporary Identifier
DFT	Discrete Fourier Transform
DL-SCH	Downlink Shared Channel
eNB	eNodeB
EPS	Evolved Packet System
FDD	Frequency Division Duplexing
FDOA	Frequency Difference of Arrival
FFT	Fast Fourier Transform
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System of Mobile Communications
HSPA	High Speed Packet Access
IDFT	Inverse Discrete Fourier Transform
IFFT	Inverse Fast Fourier Transform
IMT-Advanced	International Mobile Telecommunications Advanced
IP	Internet Protocol
ISI	Inter Symbol Interference
ITU	International Telecommunications Union
LCID	Logical Channel ID
LTE	Long Term Evolution
MAC	Medium Access Protocol

MGRS	Military Grid Reference System
MRSE	Mean Radial Spherical Error
MS	Mobile Station
OFDMA	Orthogonal Frequency Division Multiple Access
PAPR	Peak-To-Average Power Ratio
PDU	Protocol Data Unit
RACH	Random Access Channel
RAR	Ransom Access Response
RF	Radio Frequency
RFE	Radio Frequency Equipment
RSSI	Received Signal Strength Indication
SAE	System Architecture Evolution
SC-FDMA	Single Carrier Frequency Division Multiple Access
SDU	Service Data Unit
SS	Subscriber Station
SVD	Singular Value Decomposition
TA	Timing Advance
TDD	Time Division Duplexing
TDMA	Time Division Multiple Access
TDOA	Time Difference of Arrival
UE	User Equipment
UL-SCH	Uplink Shared Channel
UMTS	Universal Telecommunications System
UTRA	Universal Terrestrial Radio Access
UTRAN	Universal Terrestrial Radio Access Network
VoIP	Voice Over IP
WCDMA	Wideband Code Division Multiple Access
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access

EXECUTIVE SUMMARY

Long Term Evolution (LTE) is an emergent network technology demanding examination. Recently certified as a true Fourth Generation technology, it is poised as one of two technologies that will replace current Third Generation cellular networks as mobile subscribers use more and more data and circuit-switched networks continue to evolve into packet-switching networks using voice over Internet protocols. LTE provides a wireless high speed connection available to fixed or mobile subscribers whose location is not predetermined. Location information of subscribers on the network can be vital, especially in situations requiring emergency response teams for medical care or crisis scenarios, as well as aiding law enforcement and homeland security.

Many methods exist for geolocating radio frequency equipment and devices and each has advantages and disadvantages. Time difference-of-arrival requires receivers that are synchronized precisely, and frequency difference-of-arrival requires Doppler shifts generated by significant velocities. A solution lies within Global Positioning System (GPS) technology, where distribution of GPS chips within subscriber devices could provide location. However, this adds significant costs for manufacturing and power related issues for the mobile devices. Instead of relying on mechanisms internal to a subscriber unit to provide location, the LTE signal itself provides a network-ranging parameter called timing advance, which can be used to correlate the distance between a transmitting point, such as a tower, and the receiving mobile device entering the network.

In LTE, uplink and downlink between towers and subscriber units is conducted primarily with a frequency-division duplexing scheme, which facilitates simultaneous communication in both directions. However, the arrival of the uplink messages to the tower from multiple subscribers requires coordination to prevent interference with other devices. Proper scheduling is achieved through the use of timing advance (TA), which the base station tower sends to a device to speed up or delay its data transmission and ensure each subscriber’s data is received in the appropriate time slot.

With successful extraction of these messages from the air interface, a distance from the tower to the subscriber can be derived based on propagation speed, the speed of light, and how many units of timing advance the base station indicates to the subscriber to use. Previous studies explored this method to geolocate Global System for Mobile communications cell phones as well as for WiMAX fixed and mobile devices, and the purpose of this thesis was to apply the same techniques and principles to LTE networks, which use different physical layer parameters, modulation schemes and message formats.

Investigation of LTE specifications revealed radio frame formats and their relation with the physical layer protocols and means of transmission over the air interface. Calculations for timing advance determined a maximum possible timing advance of 0.67 milliseconds, and each unit of timing advance equal to 78.125 meters.

While laboratory testing equipment was on hand, the necessary software for analyzing LTE signals was not available, and an operational LTE testing area was still under development. Taking the calculations for timing advance derived from the LTE specifications, we employed computer software to generate simulated LTE networks with multiple base stations for testing geolocation methods.

Modeling geolocation techniques used in the two-dimensional aspect, we simulated a multiple base station network with varying scenarios of base station placements and TA-based radii fluctuations and attempted to approximate a subscriber station’s location. Results of the simulations showed that in networks with various numbers of towers and random tower placements, the location of a subscriber could generally be approximated within 60 meters provided accurate tower placement is given and timing advance offsets are known.

Taking into account that the TA-based offsets contained in the LTE signal internals corresponded to a height-generated distance from the peak of a tower to the subscriber below, we generated another set of computer simulations to examine geolocation within a three-dimensional model similar to GPS. Using TA distance as the radii for spheres radiating from a base station, where the top of the tower represents the center point of the sphere, we applied linear algebra for a spherical system of equations to

obtain a subscriber approximation generally falling within less than 50 centimeters of error provided tower characteristics of location and height and timing advance offsets are known.

Geolocation within 50 centimeters based on theoretical passive collection of LTE signal data over the air interface provides a valuable asset, both to cellular networks in need of and offering location based services, and as an aid to law enforcement, emergency response teams, and tactical personnel. Simulations for both the two-dimensional and three-dimensional methods of geolocation demonstrate this capability based on internal signal parameters.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

First and foremost, I would like to thank my wife and children; the many sacrifices I have made for my country over the past fifteen years pale in comparison to the sacrifices they have made for me. I would also like to thank the friends I have made here at the Naval Postgraduate School for their assistance in getting through the master's program as well as other obstacles over the past two years. Finally, I would like to thank my advisors, Dr. McEachen and Dr. Loomis, for their guidance throughout the thesis process.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND: LTE AND WHY WE CARE

The Third-Generation Partnership Project (3GPP) formed in 1998 acts as a standards-developing body responsible for specifications of the Third Generation (3G) Universal Terrestrial Radio Access (UTRA) and the Global System for Mobile communications (GSM) standards. The scope of the project when it was formed was to produce global specifications for a 3G mobile system based on an evolved GSM core network. The cellular technologies specified by 3GPP are the most widely deployed in the world, with more than 2.6 billion users in 2008. The latest step being studied and developed in 3GPP is an evolution of 3G into an evolved radio access, referred to as Long Term Evolution (LTE) [1].

The 3GPP intends LTE to be a fourth generation (4G) mobile-communication system that can take the telecom industry into the 2020s. In June 2005, a study item by the project finalized the requirements for LTE to ensure competiveness over a ten-year time frame. The requirements include reduced delays in terms of connection and transmission latency, increased user data rates, reduced cost per bit through improved spectral efficiency, and greater flexibility of spectrum usage in both new and pre-existing bands. Other key requirements involve simplified network architecture, reasonable power consumption for the mobile terminal, and seamless mobility, including between different radio-access technologies [2].

Release 8 of 3GPP standards provided enhancements to the existing Universal Mobile Telecommunications System (UMTS) with the specifications for LTE and the System Architecture Evolution (SAE). SAE represents the non-radio aspects of the complete LTE system and focuses on the end goal of a packet-switched core to support packet-switched radio access. This flat IP-based network architecture ensures mobility between existing mobile telecommunication systems (e.g., GSM, UTRAN, Wi-Fi, and WiMAX) and replaces the General Packet Radio Service (GPRS) core network for 2G and Wideband Code Division Multiple Access (WCDMA)-based 3G networks.

Together, LTE and SAE comprise the Evolved Packet System (EPS), where all voice and data services over the air interface are fully packet-switched versus circuit-switched.

In December 2008, 3GPP froze the Release 8 standards, allowing production of equipment based on LTE and network deployment to begin. 3GPP published Release 9, freezing its specifications in December 2009, and development of Release 10 is currently in progress. Release 10 will give the world LTE-Advanced, a further evolution of LTE that will meet the requirements of a true 4G network as outlined by the International Telecommunications Union (ITU) [3]. Assessment of LTE-Advanced against ITU's Radiocommunication Sector requirements occurred on 21 October 2010 and was accorded the official designation of an International Mobile Telecommunications-Advanced (IMT-Advanced) 4G technology [4].

Despite a lack of true 4G status, LTE networks based on the Release 8 specifications deployed around the globe. The first available networks appeared in Stockholm, Sweden and Oslo, Norway during December 2009. In North America, wireless carrier MetroPCS deployed LTE networks in Las Vegas, Nevada and Dallas/Fort Worth, Texas by the end of September 2010. Well-known carriers Verizon Wireless and AT&T embraced LTE as the future of their networks and continue to conduct testing within the United States. With the requirement of backwards compatibility for LTE-Advanced imposed by 3GPP, existing LTE networks anticipate cost-effective upgrading to the Release 10 standards in the future. The map in Figure 1 displays LTE network deployment across the globe, with red markers indicating countries that have carriers committed to establishing LTE networks and blue markers indicating actual deployment [5].

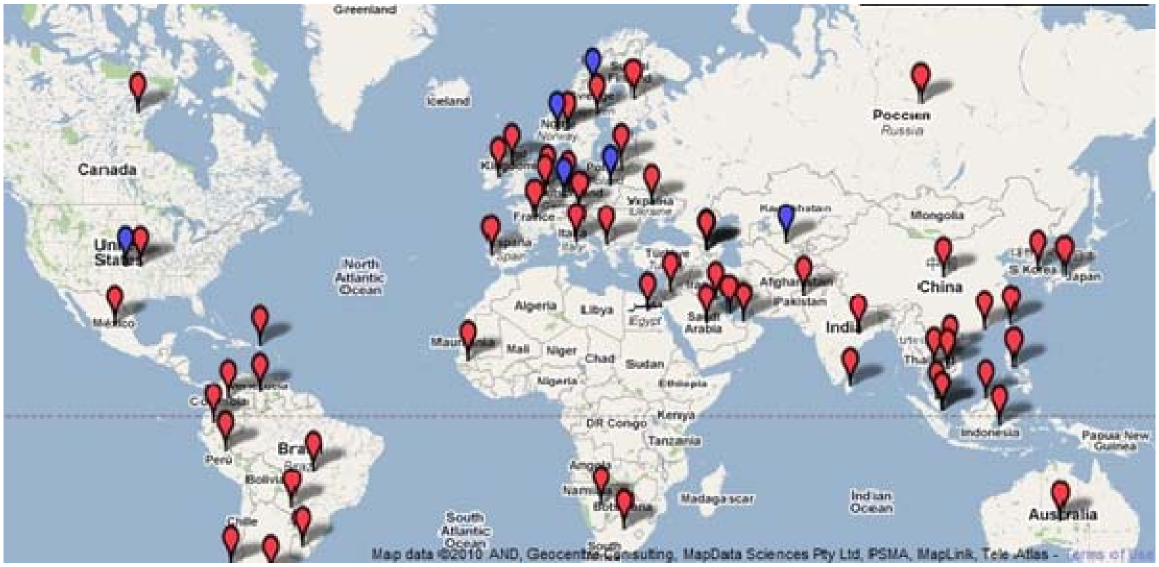


Figure 1. LTE Forum LTE Deployment map (From [5]).

LTE stands upon the brink of revolutionizing the current mobile telecommunications infrastructure. It has been embraced around the world by numerous wireless carriers as the answer to evolving their existing 3G networks and demanded attention from industrial partnerships already involved with other 4G technologies like WiMAX. This merits continued exploration of the specifications and protocols of LTE in the foreseeable future.

B. OBJECTIVE: GEOLOCATION

The ability to locate subscribers within a network remains a key element in past and future mobile telecommunication services. From an operational perspective, a mobile network would not be "mobile" if the location between subscriber stations (SS), which is synonymous with mobile stations (MS), and base stations (BS) could not identify its initial location followed by constant updating while the SS is on the move. This necessary functionality facilitates the success of location-based applications on cellular phones. Consumers use their mobile devices for navigation, weather information, traffic status and social-networking features such as identifying friends in a shared area. These services demand accurate determination of SS location in order to provide the requested information to the consumer.

Subscriber location capability proves an invaluable asset to emergency response teams and law enforcement. Situations occur where a person in distress may require immediate assistance and not have any knowledge of his or her precise whereabouts. Another situation could involve a perpetrator, sought out by police, who actively uses his or her cellular device. Both cases can have quicker outcomes provided the location of the SS can be obtained.

In 2003, Congress mandated that mobile carriers be required to provide accurate location information for the origins of 911 calls from mobile phones to allow response to such situations [6]. The FCC Public Safety and Homeland Security Bureau announced plans for a new 700 MHz public-safety radio band in August 2009 and identified LTE as the proposed technology to be used [7]. These demands validate the objective of this thesis, which is to develop a method to geolocate LTE subscribers and assess the fix accuracy that can be achieved using this technique.

C. RELATED WORK: METHODS OF GEOLOCATION

Different methods exist to provide location information on wireless devices. Many of these methods depend upon the nature of the technology employed by mobile carriers. The methods can be narrowed further when a choice must be determined between location capability built within a mobile device or reliance on external means based on the network on which the device operates.

A global positioning system (GPS) chip represents an example of an internal capability solution for geolocation. The addition of a GPS chip in a wireless handset can provide accurate location information as well as satisfy the Congressional requirement concerning 911 emergency calls from the device. As reliable as GPS has been proven to be, there remain significant drawbacks to its use with a wireless handset. Higher rates of power consumption would occur due to the necessary transmission increase of data coupled with energy requirements for the GPS chip itself. Another drawback lies in the cost increase to manufacture mobile devices with onboard GPS. Disabling or failure of the GPS on the handset could lead to detrimental situations, especially during a 911 response or a law-enforcement pursuit where accurate location is vital. Mitigating these drawbacks is possible through use of external approaches to geolocation. While they may not provide location results as precise as GPS, external methods overcome the obstacles presented with power, cost, and bandwidth. External methods also ensure redundancy by continuing to provide location information should the GPS capability fail.

Several possible passive external techniques exist to locate a radio frequency (RF) device, including received-signal-strength indication (RSSI), angle-of-arrival (AOA), time-difference-of-arrival (TDOA), frequency-difference-of-arrival (FDOA), or potential internal signal characteristics [8],[9]. Multipath and variable broadcast strength pose limits on reliable location acquisition when directly applying RSSI. FDOA provides accurate location for platforms such as aircraft but requires significant relative motion to generate Doppler shift. Therefore, it is inadequate for terrestrial geolocation. For locating an LTE subscriber, implementing acquisition schemes based on AOA, TDOA, and signal internals remains the best possibility.

Existing mobile carriers have already proven means of geolocation for their specific network technologies based on AOA, TDOA, and signal internals or a combination of them. The nationwide time-division multiple access (TDMA) and GSM mobile carriers use network-based location schemes. Access to the shared spectrum is controlled by timing data built into the structure of TDMA and GSM signals, allowing carriers like T-Mobile and AT&T Mobility to use TDOA methods. Providers using code division multiple access (CDMA), such as Verizon Wireless and Sprint, use an assisted GPS method with their mobile devices.

GSM provides an internal parameter to TOA within its signal characteristics known as timing advance (TA). Timing advance is used by the network and handset to align the traffic bursts with the TDMA frames of GSM. Using this timing advance and speed of propagation, we can approximate range rings from base station towers, and the intersection of these range rings from multiple towers provides an approximate location for the mobile station [10]. An illustration of overlapping timing advance range rings is shown in Figure 2 [8]. Further refinement of this approach found averaging multiple timing advance measurements minimized error in random variable sampling, tightening location accuracy [11].

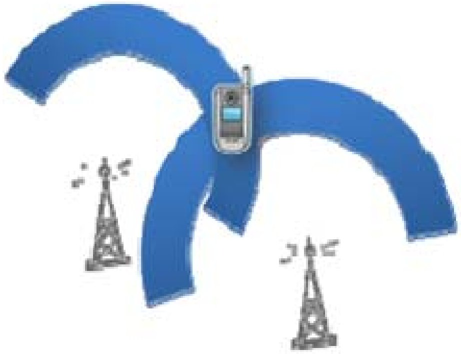


Figure 2. Illustration of overlapping timing advance range rings (From [8]).

Similarities to this signal characteristic exist in the IEEE 802.16 standards for Worldwide Interoperability for Microwave Access (WiMAX), a technology competing with LTE for 4G status. The uplink in the 802.16 medium access control (MAC) is also shared between SSs in a TDMA fashion, with initial assignment of timing adjust generated by the BS after the initial entry and ranging request by a SS [12]. An investigation of the same approach, using radial distances from multiple BS based on the WiMAX timing adjust, yielded the possibility of location fixes with 10 times greater accuracy than previous results in literature applying timing advance techniques to GSM networks [8].

D. APPROACH

3GPP incorporates a timing advance function similar to GSM and timing adjust in WiMAX. If the timing of a specific terminal needs correction, the network issues a timing advance command for this specific mobile terminal, instructing it to retard or advance its timing relative to current uplink timing. The user-specific timing advance command is transmitted as a MAC control element on the downlink-shared channel (DL-SCH) [1]. The approach to geolocation in this thesis uses the same location-deriving principles employed through previous WiMAX research; knowledge of the signal internals in relation to timing data in order to isolate subscriber station position based on radial distances to known base station locations is used.

First, investigation of the 3GPP standards was conducted to establish where the timing data resides within the signal internals, what packets contain the timing data, and how the timing data is calculated. The results of this investigation were compared to other sources for validity and correctness. Due to a lack of testing equipment software and field-testing facilities with an established LTE network, computer simulation was used to test TA-constrained behavior with multiple base stations under variable conditions, such as random distances and angled placement. Accuracy of location acquisition based on the TA ranging parameter in LTE networks was established based on the results of these computer simulations.

E. THESIS ORGANIZATION

Investigation of the 3GPP standards for LTE begins in Chapter II. Exploration of the physical channels and modulation, physical layer procedures, and initial network entry access identify proper extraction of TA data. The methods of approximation to establish geolocation with supporting mathematics are described in Chapter III, and computer simulation based on a two-dimensional model with variable constraints is conducted. The simulations use a Monte Carlo test scheme to establish fix accuracy in real-world LTE networks.

After exploration of a two-dimensional geolocation method, the three-dimensional problem with TA, where the extracted TA value correlates to an angled distance between the subscriber station and the base station, is addressed in Chapter IV. Methods of approximation and calculations through mathematics are discussed, followed with Monte Carlo computer simulations that encompass three-dimensional aspects. Finally, conclusions on the overall simulation results and recommendations for furthering and refining this research are discussed in Chapter V.

THIS PAGE INTENTIONALLY LEFT BLANK

II. INVESTIGATION OF LTE WORKINGS

LTE intends to deliver superior performance compared to existing 3GPP networks based on High Speed Packet Access (HSPA) technology. Performance targets based on peak user throughput should be minimum 100 Mbps in downlink and 50 Mbps uplink, which is ten times more than previous released HSPA technology from 3GPP. With this capability, LTE proves to be a more than suitable platform for Internet protocol (IP) applications at the upper levels of the protocol stack such as large data transfers and voice over IP (VoIP). Other performance targets with LTE include transmit round trip time less than 10 ms, optimized terminal power efficiency, and bandwidth flexibility ranging from below 1.5 MHz up to 20 MHz allocations. To understand how these performance targets are achieved and why a timing advance feature is necessary, an investigation into the inner workings of LTE is required.

A. UPLINK AND DOWNLINK

LTE relies upon multiple access schemes that allow several subscriber stations, referred in the 3GPP standards and throughout the remainder of this thesis as user equipment (UE), to share the capacity of the network. Contrary to other existing mobile technologies, LTE uses different multiple access schemes for uplink and downlink. They are single-carrier frequency-division multiple access (SC-FDMA) and orthogonal frequency-division multiple access (OFDMA), respectively.

OFDMA operates on the same principles as traditional frequency division multiplexing. Different frequencies can carry different pieces of information. A good portrayal of the concept lies with conventional radio, where different stations offer different music on different frequencies. A digital modulation technique called orthogonal frequency-division multiplexing (OFDM) allows a radio station to use a group of frequencies rather than a single carrier. Allotted bandwidth is subdivided into spaced frequencies to create a sub-channel. These sub-channels allow simultaneous transmission of data in the form of multiple symbols combined via inverse fast Fourier transform (IFFT). Closely-spaced orthogonal sub-carriers are then used to carry the sub-channels, one for each carrier, and achieve significant avoidance of inter symbol interference (ISI). To accomplish OFDMA, different OFDM sub-channels are assigned to different users, improving upon the avoidance of ISI and increasing overall network capacity. Parameters for the downlink transmission scheme, including transmission bandwidth, fast Fourier transform (FFT) sizes and sampling frequencies, is included in Appendix A [13].

There are disadvantages to using OFDM, primarily attributed to the use of allotted bandwidth. When tightly packing many carriers in an allotted bandwidth, these narrower frequency bands become wider in time. A symbol that is wider in time provides greater protection against time-smearing effects of multipath propagation, where the signal is received as different reflections and is subject to fading. However, the frequency bands suffer from increased sensitivity to Doppler shift as they become narrower and may result in nulls failing to align, thereby creating ISI. Another disadvantage of OFDM and OFDMA concerns power requirements, since its use generates high peak-to-average-power ratio (PAPR). There is little concern for PAPR when discussing the downlink side, since an LTE base station, referred in the 3GPP standards and throughout the remainder of this thesis as an eNodeB (eNB), can easily receive sufficient power. Rather, the problem lies on the uplink side for the UE. In order to compensate for PAPR while taking advantage of OFDMA capability, 3GPP instituted SC-FDMA as the multiple access scheme for uplink from UE.

SC-FDMA provides linearity to a pre-coded OFDMA scheme by adding an additional discrete Fourier transform (DFT) before the conventional OFDMA processing. Multiple access among users requires assignment of different sets of nonoverlapping, silent Fourier coefficients at the transmitter prior to the initial IFFT for the OFDMA signal. The OFDMA signal now becomes a single-carrier transmit signal and no longer requires modulation of multiple sub-carriers by the transmit symbols. The silent Fourier coefficients are then removed on the receiver side after the signal runs through the FFT. By adding the extra DFT, SC-FDMA reduces the PAPR of the signal. This allows for cost-efficient terminal power amplifier design for the UE and directly expands UE battery life.

An overview of the uplink and downlink schemes described and the functional commonality between the SC-FDMA and OFDMA transmit chain is shown in Figure 3 [14], with white blocks common to both multiple access schemes and darkened blocks specific to SC-FDMA only. The constellation mapper converts an incoming bit stream to single carrier symbols through modulation. Then the time domain single carrier symbols are converted from a serial format to parallel symbol blocks. Next, additional DFT combines with the time domain SC symbol blocks to form M discrete frequency tones. Output tones for SC-FDMA or the original bit stream for OFDMA are mapped to specified subcarriers and applied to an inverse discrete Fourier transform (IDFT). At this stage, optimal spacing of the orthogonal subcarriers via a cyclic prefix (CP) is necessary to avoid ISI, and pulse shaping prevents spectral regrowth that may occur. Radio Frequency Equipment (RFE) converts the digital signal to an analog signal, and the signal is transmitted to the destination RFE receiver. For the receive side of the chain, the process is essentially reversed, with the end result of the original bit stream.

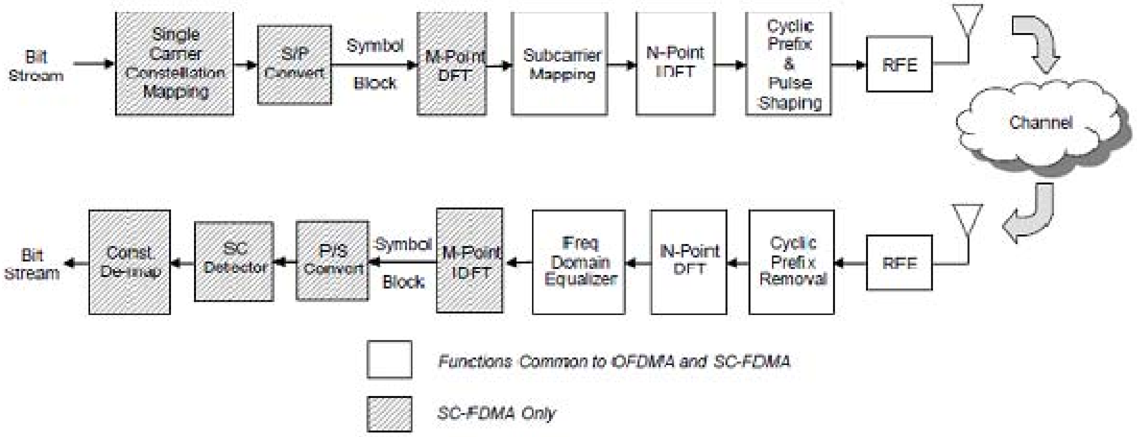


Figure 3. Functional commonality between SC-FDMA and OFDMA signal chains (From [14]).

LTE provides two modes of operation for uplink and downlink sharing the same frequency band for transmission and reception: frequency-division duplexing (FDD) and time-division duplexing (TDD). The major difference between the two schemes is the use of paired and unpaired frequency bands. FDD uses different frequencies on paired bands for continuous UE transmission and reception with reasonable separation between downlink and uplink directions. TDD uses the same frequency band for transmission and reception but alternates the transmission direction in time with a guard period for scheduling. How uplink and downlink are coordinated with the different schemes is illustrated in Figure 4 [14]. LTE eNodeBs can implement either scheme, but FDD is preferred because it provides the maximum achievable data rates. The specifications for LTE include FDD and TDD in all of its descriptions since there is little to no difference in the physical architecture of the technology, and the procedures are the same. The FDD mode of operation is the focus of this thesis.

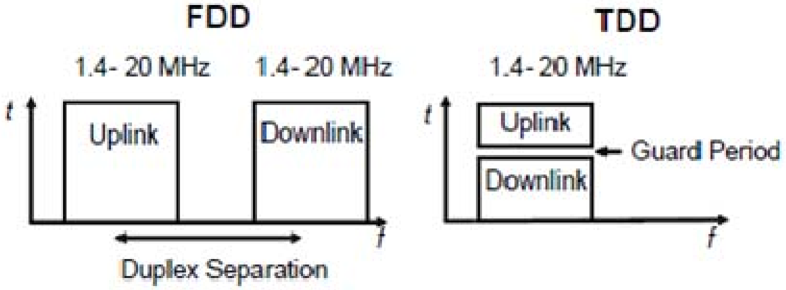


Figure 4. Principles of TDD and FDD modes of operation (From [14]).

With eNodeBs providing simultaneous transmission and reception with FDD, it comes into question why a timing advance feature is required. The timing control procedure is needed so that the uplink transmissions from different users arrive at the eNodeB essentially within the cyclic prefix. Such uplink synchronization is needed to avoid interference between the users with uplink transmissions scheduled on the same subframe. The eNodeB continuously measures the timing of the UE uplink signal and adjusts the uplink transmission timing as shown in Figure 5 [14]. Timing advance commands are sent via random access channel (RACH) only when a timing adjustment is actually needed, i.e., the uplink for the UE is not synchronized. This occurs during system access, periods of UE idleness or inactivity, non-synchronized handover from eNodeB to eNodeB, and radio link failure. To better realize how these situations occur, investigation of the LTE specifications for radio frame structure and timing was conducted.

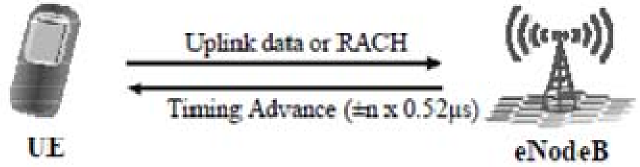


Figure 5. Uplink timing control (From [14]).

B. RADIO FRAME STRUCTURES AND TIMING

3GPP specification TS 36.211 *Physical Channels and Modulation* describes standard radio frame structures and timing and how they are implemented in LTE for uplink and downlink. A standard LTE unit of time used throughout all the specifications is given to describe the size of various fields in the time domain, and is expressed as

$$T_S = 1/(15000 \times 2048)\,\text{sec}. \qquad (1)$$

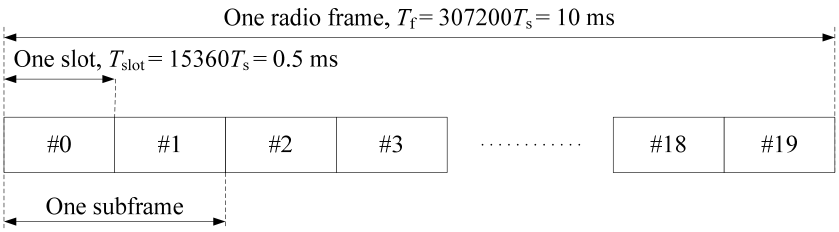
In Equation (1), 15000 represents the 15 kHz sub-carrier spacing and 2048 is the maximum FFT size. Downlink and uplink transmissions are further organized into radio frame structures with duration of T_S multiplied by the maximum sampling frequency of 30.72 MHz. This is expressed as

$$T_f = 307200 \times T_S = 10\,\text{ms}. \qquad (2)$$

Radio frame structures are further divided by slots. These slots are expressed in the time domain as

$$T_{slot} = 15360 \times T_S = 0.5\,\text{ms}. \qquad (3)$$

Two radio frame structures are supported in LTE, with type 1 applicable to FDD. Each radio frame is $T_f = 10$ ms long and consists of 20 slots of length $T_{slot} = 0.5$ ms, numbered from 0 to 19. Subframes within a frame is defined as two consecutive slots, where subframe *i* consists of slots 2*i* and 2*i*+1. The type 1 radio frame structure is illustrated in Figure 6 [15].



One radio frame, $T_f = 307200T_s = 10$ ms
One slot, $T_{slot} = 15360T_s = 0.5$ ms
#0
#1
#2
#3
#18
#19
One subframe

Figure 6. Frame structure type 1 (From [15]).

TS 36.211 gives an overall description of uplink-downlink frame timing and calculates the start time from $(N_{TA} + N_{TA\ offset})T_S$. Transmission of the uplink radio frame i from the UE is to start $(N_{TA} + N_{TA\ offset})T_S$ seconds before the start of the corresponding downlink radio frame at the UE, where $0 \le N_{TA} \le 20512$, and $N_{TA\ offset} = 0$ for frame structure type 1 [15]. The uplink-downlink timing relation is demonstrated in Figure 7 [15]. As for N_{TA}, this corresponds to the number of timing advance units that need to be applied to the start times of the uplink and downlink radio frames and gives us a maximum of 20512 TA units. The assignment of TA units occurs in the physical layer of LTE through use of MAC protocols.

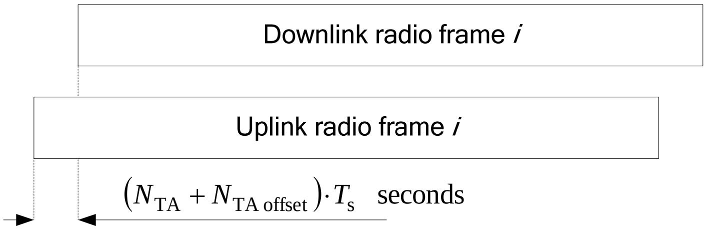


Figure 7. Uplink-downlink timing relation (From [15]).

C. MEDIUM ACCESS CONTROL

3GPP TS 36.321 *Medium Access Control (MAC) protocol specification* gives a functional point of view for the MAC architecture used in LTE. The MAC provides numerous functions to support error correction, scheduling information reporting, and priority handling between UEs by means of dynamic scheduling. This latter function is instrumental for solving synchronization situations as previously discussed, where adjustment of frame timing for a UE is required to join or rejoin the network.

A MAC protocol data unit (PDU) is a bit string that is byte aligned in length and consists of a MAC header, zero or more MAC Service Data Units (SDU), zero or more MAC control elements, and optional padding. The MAC PDU header is further divided into one or more MAC PDU subheaders. An example of a MAC PDU is portrayed in Figure 8 [16].

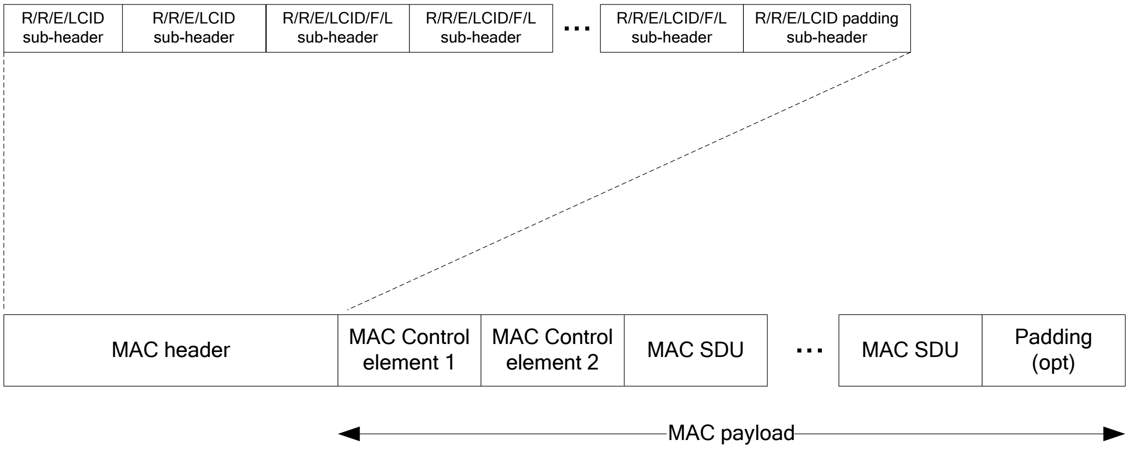


Figure 8. Example of MAC PDU consisting of MAC header, MAC control elements, MAC SDUs and padding (From [16]).

In cases where a UE uses the RACH for joining or rejoining the network, an eNodeB responds with a MAC PDU that substitutes MAC Random Access Responses (RAR) as the MAC control elements to synchronize the UE. This variation of the MAC PDU is illustrated in Figure 9 [16].

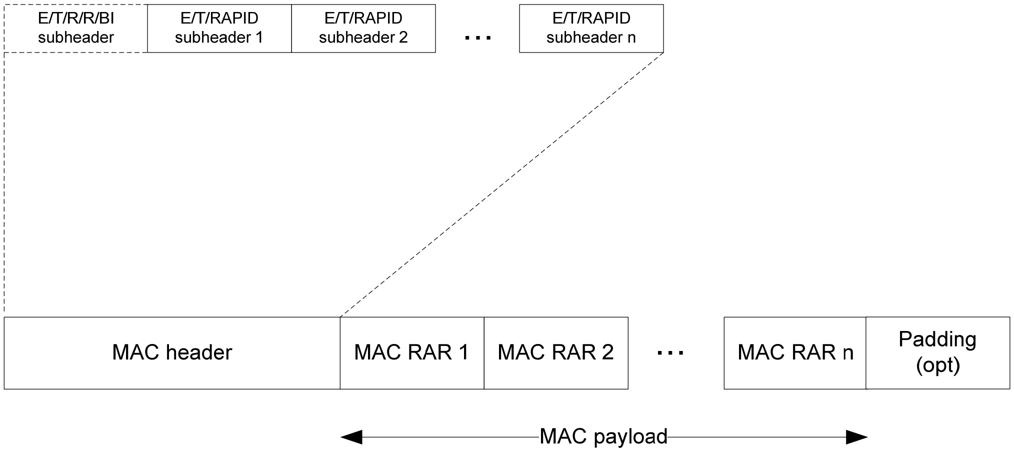


Figure 9. Example of MAC PDU consisting of MAC RARs (From [16]).

The MAC RAR, displayed in Figure 10 [16], consists of six octets divided into four fields: a reserve field of 1 bit set to 0, Timing Advance Command, an Uplink Grant field to indicate resources available to the UE, and a Temporary Cell Radio Network Temporary Identifier (C-RNTI). The timing advance command field is used to indicate the amount of timing adjustment that UE has to apply. The field comprises bits [6:0] in the first octet of the MAC RAR and bits [7:4] of the second octet, for a total of 11 bits. The specifications define these 11 bits to indicate the index value of T_A, ranging from 0 to 1282. Therefore, the maximum T_A is represented by a binary 10100000010 for decimal 1282.

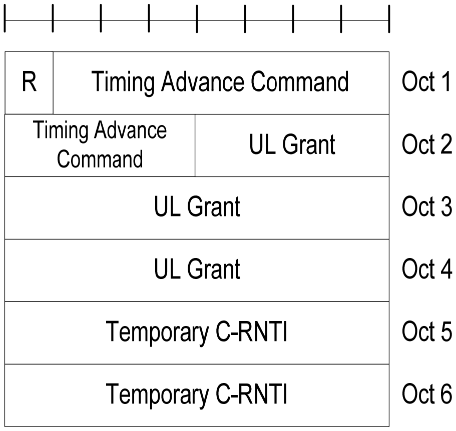


Figure 10. MAC RAR (From [16]).

After the UE successfully joins or rejoins the network, communication between the UE and the eNodeB moves from the RACH and now resides on the uplink shared channel (UL-SCH) and DL-SCH. As the UE changes distance between itself and the eNodeB, timing advance may need to be adjusted from its initial indication in the MAC RAR to compensate. The timing advance control element is identified by the MAC PDU subheader with the logical channel ID (LCID) field shown in Figure 8. This field, illustrated in Figure 11 [16], is one octet long, with the two most significant bits as reserved bits set to 0, and the remaining 6 bits to indicate index values of T_A adjustment ranging from 0 to 63.

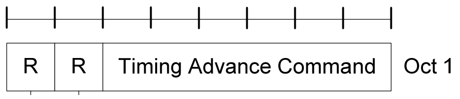


Figure 11. Timing Advance Command MAC control element (From [16]).

D. TIMING ADVANCE CALCULATIONS

3GPP TS 36.213 *Physical layer procedures* [17] give further refinement on transmission timing adjustments, where the timing advance command indicates the change of the uplink timing relative to the current uplink timing as multiples of $16T_S$. In case of RAR, 11-bit timing advance command T_A indicates N_{TA} values by index values of T_A = 0, 1, 2, …, 1282, where an amount of the time alignment is given by $N_{TA} = T_A \times 16$. In other cases, 6-bit timing advance command T_A indicates adjustment of the current N_{TA} value $N_{TA,old}$ to the new T_A value $N_{TA,new}$ by index values of T_A = 0, 1, 2, …, 63, where $N_{TA,new} = N_{TA,old} + (T_A - 31) \times 16$. Here, adjustment of N_{TA} by a positive or negative amount indicates advancing or delaying the uplink transmission timing by a given amount, respectively. The specification goes on to describe how timing advance is applied to the 10 ms radio frame in Figure 6. For a timing advance command received on subframe *n*, the corresponding adjustment of the timing applies from the beginning of subframe *n+6*.

With all the necessary parts of timing advance in LTE explained, calculation of time and distance with respect to one unit of TA can be achieved. Review of the literature suggests a common statement with the maximum $T_A = 0.67$ ms correlating to a radial distance just over 100 km from the eNodeB to the UE. To verify this statement, knowledge of the maximum index values for T_A and employment of (4) were used. For example, an RAR message is received with the maximum index value $T_A = 1282$. Application of this to N_{TA} yields $N_{TA} = T_A \times 16 = 1282 \times 16 = 20512$, the maximum N_{TA} for radio frame type 1. Substituting this value in (4), we get the expected maximum result from literature [1]

$$(N_{TA} + N_{TA\ offset}) \times T_S = (20512 + 0) \times 1/(15000 \times 2048) \approx 0.67 \text{ ms.} \tag{4}$$

The distance in meters per unit of TA can be derived with knowledge of the sampling frequency in LTE. Sampling frequencies for different transmission bandwidths in LTE are indicated in Table 1 as multiples of 3.84 MHz. By multiplying the reciprocal of the sampling frequency$1/F_S$ by the speed of light, approximately 3×10^8 meters per second, each unit of TA should correlate to a distance of approximately 78.125 meters. In regard to the example above, a maximum index value of 1282 units of TA multiplied by 78.125 meters produces 100.156 km, validating the assertion found in literature.

The specifics of LTE network communication was investigated in this chapter and specific parameters of interest were identified. Initial joining or rejoining of a UE to the network provides responses from the eNodeB with timing adjustment instructions based on distances between them and further adjustments based on the change in distance due to UE mobility. These instructions are facilitated through MAC protocol internals as deemed necessary by the eNodeB. TA is able to be resolved to distance and allows for a radial distance from the eNodeB to be calculated. This distance can be applied to geolocation with use of intersecting radii, which was introduced in the previous chapter. With detailed knowledge of bit specifics from the standards, computer simulations to provide estimates of geolocation accuracy in variable networks with multiple eNodeBs are documented in the next chapter.

III. TWO-DIMENSIONAL SIMULATIONS

A. METHOD OF APPROXIMATIONS

The initial approach to simulating geolocation of an LTE UE requires establishment of parameters to be used. First, the simulation employs a flat Earth model with calculations derived in Cartesian coordinates. This provides for a meter-by-meter coordinate system with simplified calculations of range radii, intercept points and probability polygons. Rather than explore a spherical model with latitudes and longitudes, the typical error over the geode due to curvature in this flat Earth model remains small, even after mapping the results to the Earth's surface with a coordinate transformation.

Second, the method of intersecting radii is based upon propagation delay acquired from signal internals. Assuming free space propagation at the speed of light, approximately 3×10^8 meters per second, at the bandwidth used, each unit of TA increases the range radius from the eNodeB by 78.125 meters, as discussed in Chapter II. Calculations of range radii assume a best case scenario with an absolute TA amount of 78.125 with no variance and deviation in the measurement.

The third parameter offsets the best case scenario by introducing deviation into the range radii calculations. Investigation into the LTE signal internals to this point have been limited to values and calculations derived from the specifications. As previously discussed, a lack of LTE-capable signal analysis equipment and field testing facilities prevented any real-world, real-time collection of TA data. Regardless of a high degree of confidence in the calculations for TA, it would be unrealistic to assume there would be no deviation from the mean 78.125 meters when analyzing an LTE signal retrieved from the air. Since it was not possible to establish a real-world value for deviation or blindly accept that there would be deviation present, the simulation required multiple runs.

The first run maintained a mean value of TA with 78.125 meters as derived from calculations. The second run introduced two forms of standard deviation based on measurement error and quantization error.

Measurement error used in the simulations is based on the results of previous research on geolocation with WiMAX. In this research, field collection results of successful TA data extracted from the air interface were combined into a single probability density function (PDF) and established an overall standard deviation of TA to be 0.673 meters [8]. To mimic this finding, a randomly selected standard deviation of 0.5 meters is used to introduce measurement error per unit of TA.

Quantization error is based on the radii from each eNodeB consisting of multiples of TA equal to a uniform distribution of ±78.125 meters. Therefore, a change in the radius Δr will have a range of $-78.125/2 < 0 < 78.125/2$ meters, as depicted in Figure 12. The probability of Δr is

$$P(\Delta r) = \frac{1}{78.125}. \tag{5}$$

Solving for the variance, we have

$$\begin{aligned} \sigma^2 = E\left[(\Delta r - 0)^2\right] &= 2\int_0^{78.125/2} \Delta r^2 \times P(\Delta r)dx \\ &= \frac{2}{78.125} \times \frac{1}{3}\left(\frac{78.125}{2}\right)^3 \\ &= 508.6\ . \end{aligned} \tag{6}$$

From this variance, a standard deviation of TA due to quantization error can be established as $\sigma = 22.5$ meters. For simulation purposes, a quantization error equal to TA, 78.125 meters, is multiplied by a uniformly distributed random number between zero and one and then added to calculated radii. A bias equal to half the quantization error, 39.0625 meters, is subtracted from the end radii result.

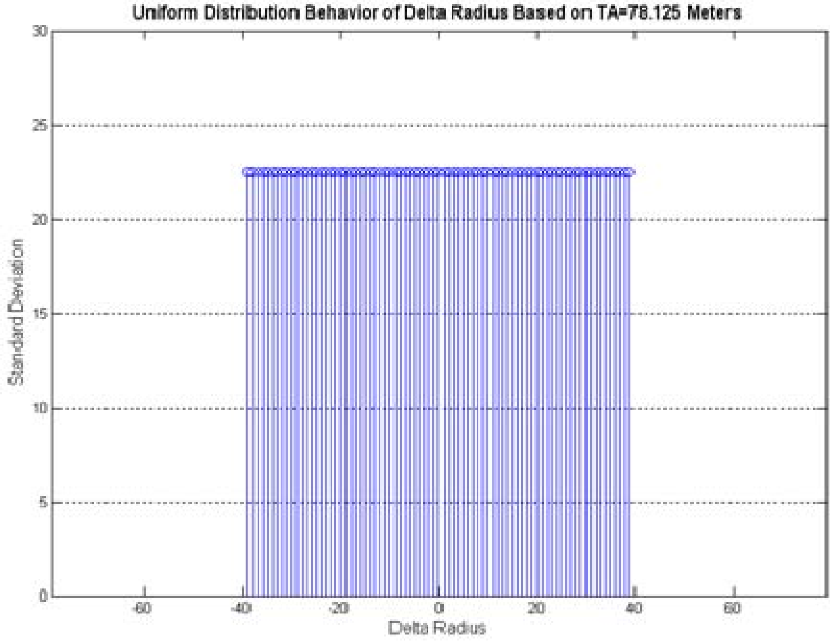


Figure 12. Uniform distribution of Δr .

B. LIKELY LOCATION CALCULATIONS

With two eNodeBs having a known generated radius, the most likely location of a UE would lie where the radius rings overlap or the point where they are closest to overlapping. Three situations must be accounted for: two radius circles can be separated without touching, one radius circle can be completely contained in the other, or the radius circles intersect. Overlapping radius circles generated from two eNodeBs results in two points of intersection, with only one of the points representing the true coordinate solution to the most likely UE location. The addition of a third eNodeB will almost always remove the ambiguity between the points of intersection based on the fact that all three radius circles only converge near one of the two intersections from the two eNodeB solution. The decision process between the three possible situations is illustrated in Figure 13.

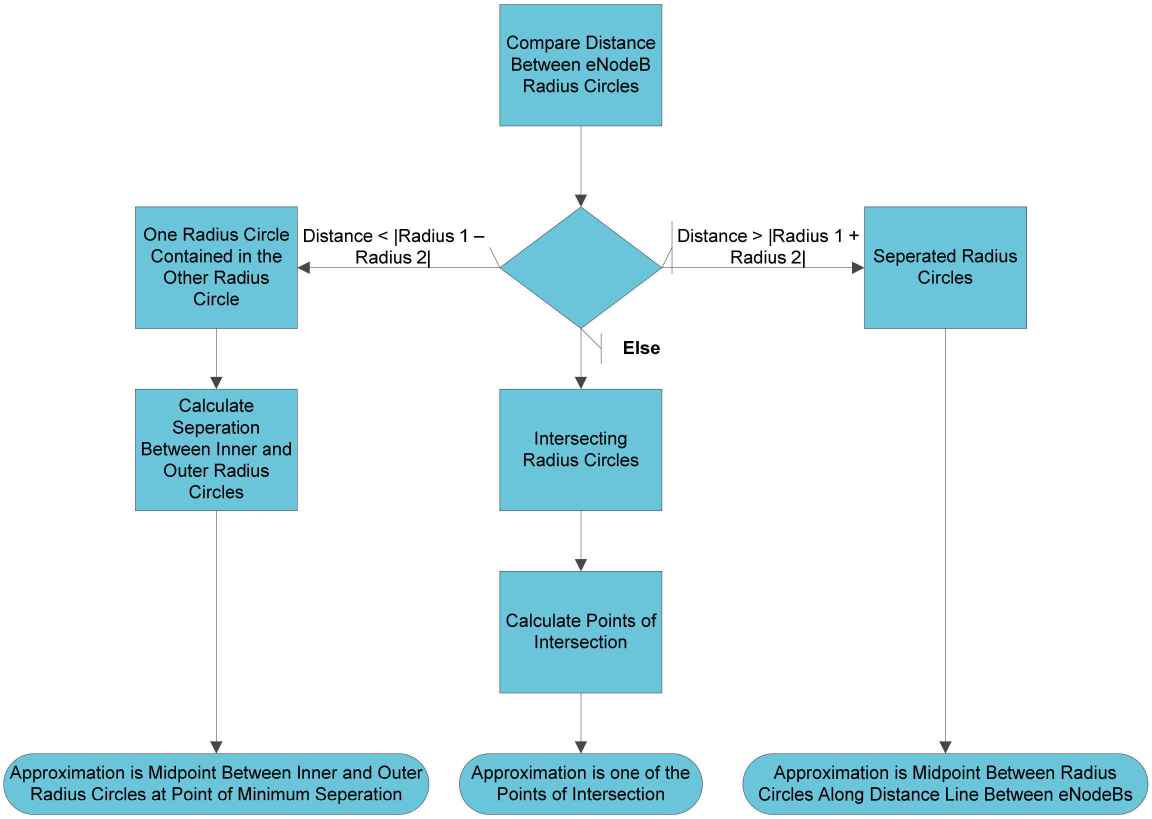

Compare Distance Between eNodeB Radius Circles
One Radius Circle Contained in the Other Radius Circle
Distance < |Radius 1 – Radius 2|
Distance > |Radius 1 + Radius 2|
Seperated Radius Circles
Else
Calculate Seperation Between Inner and Outer Radius Circles
Intersecting Radius Circles
Calculate Points of Intersection
Approximation is Midpoint Between Inner and Outer Radius Circles at Point of Minimum Seperation
Approximation is one of the Points of Intersection
Approximation is Midpoint Between Radius Circles Along Distance Line Between eNodeBs

Figure 13. Decision process for geolocation method of intersecting radii.

For two eNodeBs generating intersecting radii, the best possible location approximation occurs when the two points of intersection get closer and closer together, eventually converging on one point. Radius circles that are barely touching or separated by a very small distance can still provide an accurate approximation despite the fact that they do not intersect. However, the increase in distance between nonintersecting radii is directly proportional to the amount of error in the approximation, making the estimate less and less meaningful as the radii become farther apart.

Using the decision process displayed in Figure 13, we obtain the distance between the eNodeBs using the Pythagorean Theorem based on their X-Y plane coordinates and compared to the sum and difference of their radii. The result of the comparison determines which of the three situations has occurred. If the sum of the radius circles is less than the total distance between the two eNodeBs, then a case of nonintersecting radii has occurred. Therefore, the approximated UE location becomes the midpoint of the total distance plus an approximation radius based on a scaling factor and the distance between the two radius circles along the same line of distance.

When the absolute value of the difference of the radius circles equates to something greater than the total distance between the eNodeBs, one radius circle is completely contained within the other. As long as the eNodeBs are not located on the same point, preventing any indication of direction between the two, drawing a straight line through the sites on both sides will provide the maximum and minimum separation of the radii. The approximated UE location becomes the midpoint between the inner and outer radius circles at the point of minimum separation.

If the radius circles are not completely separate or contained one within another, then they intersect. There is a greater chance of two points of intersection and rarely will the radius circles intersect on one specific point. The most accurate method of approximating the UE location in this case is with triangles rather than mapping all points along both radius circles in an attempt to find the closest pair. The use of triangles to calculate the intersections of circles is illustrated in Figure 14.

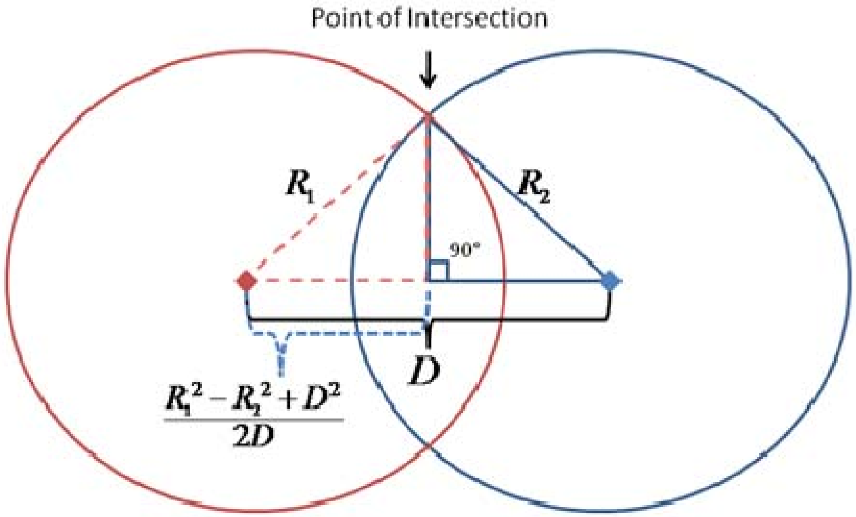


Figure 14. Use of triangles to calculate intersection of circles.

Based on the radii generated from the eNodeBs and the total distance between them, the law of cosines can be used to determine the distance to the middle of the overlapping portions of the radius circles. Derivation of this distance requires application of the Pythagorean Theorem with the given information already known, such as each hypotenuse corresponds to each radius and each triangle base, or adjacent side, is the total distance between the eNodeBs minus the distance from the midpoint to the opposite eNodeB. Both triangles share the last, or opposite, side. This yields a solution for both triangles for the shared side. The two equations are equated and solved for the distance from either eNodeB to the midpoint.

Knowledge of the midpoint provides a line through the two intercept points that is perpendicular to the distance line between the eNodeBs. With the derived base distance and the hypotenuse, the triangles can now be solved completely to find the distance from the midpoint to the intercept that was previously eliminated when the two equations were equated. It is now feasible to calculate the intersect points relative to the midpoint or the corresponding eNodeB.

To elaborate in terms of Cartesian coordinates, an eNodeB at the origin and another eNodeB a distance D along the X-axis, the X-coordinate for both intersection points is calculated using the inter-site distance D and the radii R from

$$X = \frac{R_1^2 - R_2^2 + D^2}{2D}. \tag{7}$$

With Pythagorean Theorem, the Y-coordinates for the intersections are derived as:

$$Y = \pm\sqrt{R_1^2 - \left(\frac{R_1^2 - R_2^2 + D^2}{2D}\right)^2}\ . \tag{8}$$

The case of intersecting radii forms probability ellipses in one of two ways, either overlapping radii from within due to the centers of both circles contained in the larger radius, or two circles overlapping from the outside. Overlapping radii from the outside gives a more reasonable approximation when dealing with only two eNodeBs. In both situations, the approximation lies on the midpoint of the ellipse, where the major axis is the distance between the intercepts and the minor axis is the distance between the radius circles.

To demonstrate the potential for geolocating a UE within a real world LTE network, software-implemented simulations were produced using MATLAB. Multiple eNodeB scenarios were simulated with the mean distance per unit of TA of 78.125 meters derived from calculations. The same scenarios were then executed a second time with the application of standard deviation and compared against the first run.

C. TWO BASE STATION SIMULATION

The first simulation uses the algorithm presented with two eNodeB radii to approximate the location of a UE. In all cases, the UE's true location is the origin on the X-Y plane. The simulation creates two eNodeBs at varying degrees in relation to the UE, each with normally distributed random distance with a mean of one kilometer from the UE, standard deviation within a range of 300 meters, and TA per the calculated value of 78.125 meters. Since the value of TA is discrete, the eNodeB rounds the radial distance to a whole unit of TA. A Monte Carlo simulation of one hundred thousand iterations was conducted for each increment of ten degree angles from 0 degrees to 180 degrees. The average distance from the center of the approximation polygon, i.e., the midpoint between the two radius circle intersections, to the actual UE location for each separate run was recorded.

The most accurate approximation of the UE location occurs at 180 degrees when the placement of the eNodeBs and the UE form a straight line with one another. As the angle decreases, the approximation error increases due to the two intersections of the radii becoming farther and farther apart. The ambiguity as to which intersection represents the UE location causes the approximation midpoint between the intersections to increase in distance from the actual UE. It remains true that one of the intersections is always close to the actual UE location, but the algorithm itself cannot differentiate between the two without some outside-assisted prior knowledge. There comes a point in the simulation where, as the angle continues to decrease, the approximation error stabilizes and begins to decrease. This is due to the angle between the eNodeBs approaching zero, and the UE and eNodeBs are again in a line with each other. The overall approximation error results of the two eNodeB simulation are displayed in Figure 15.

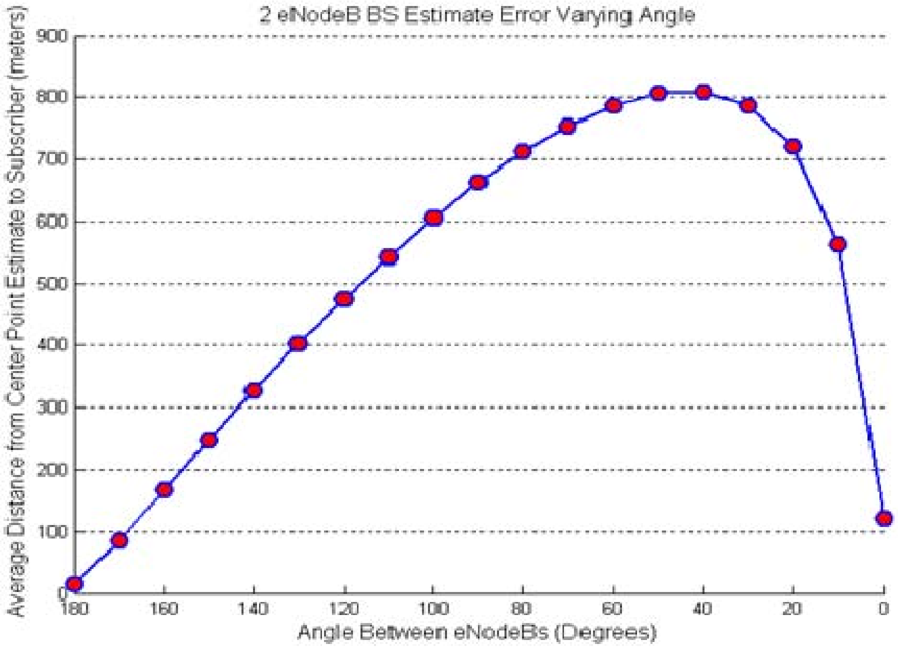


Figure 15. Approximation error analysis of two eNodeBs varying angle simulation.

The simulation was repeated with the addition of TA standard deviation from measurement error ranging to 0.5 meters and quantization error ranging to 78.125 meters. The intent was to introduce fluctuations in the TA measurements and required a modification to the radii derivation in the MATLAB code. The eNodeB rounds the radial distance to a whole unit of TA as before and then adds a normally distributed random error factor based on the 0.5 meter standard deviation. Once the site radii is calculated, a quantization error of 78.125 meters multiplied by a uniform randomly distributed number between zero and one is added to the radii length, followed by an additional bias of half the quantization error, 39.0625 meters, which is subtracted from the end result. This creates a higher percentage of radii that are greater in distance than the previous random radii generated in the first run. As a result, the approximation error increased slightly.

However, the approximation error followed the same line of distribution as the last simulation despite the addition of standard deviation. The approximation error results with standard deviation applied are shown in Figure 16. Sample plots generated during the two eNodeB simulation, showing results at varying angles between the eNodeBs, are contained in Appendix B.

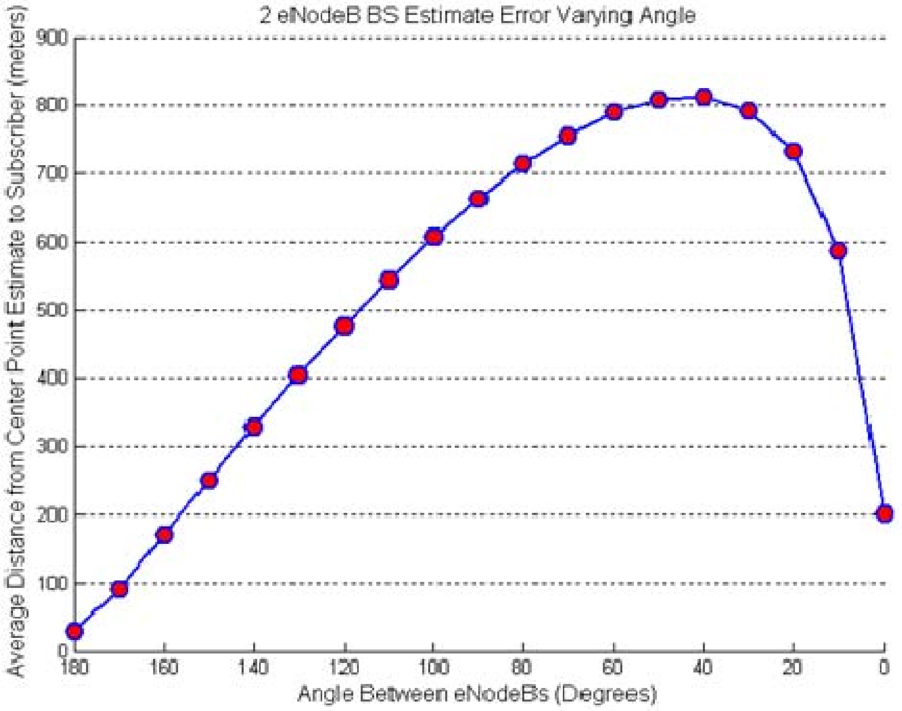


Figure 16. Approximation error analysis of two eNodeBs varying angle simulation, standard deviation applied.

D. MULTIPLE BASE STATION SIMULATION

The second set of simulations conducted in MATLAB explores geolocating a UE with more than two eNodeBs available. The introduction of a third or more eNodeB radius circle offsets the limitations presented in a two eNodeB approximation estimate by removing the ambiguity between the two points of intersection and their relation to the actual UE position. Thus, a more accurate position estimate on the actual UE location can be derived.

The approach to the multiple eNodeB simulation uses the same algorithm presented in the previous simulation. The intersections of each eNodeB radii are calculated and each intersection compared to the closest intersection of the next pair of radii. Once found, the closest intersection coordinate points are added to an array, and the process repeats for the remaining pairs. The completed array of closest intersections forms the vertices for a small polygon correlating to the approximation point of the UE. Using a three-eNodeB network as an example, we derived three pairs of intersections for a total of six points of intersection. The three closest intersection points are chosen, which when plotted should form a small triangle near the true UE location. The exact shape and number of vertices of this polygon is unknown. Therefore, to calculate the approximation point, the coordinate values of X and Y for the chosen intersections are averaged together, respectively, to find the approximation point coordinates on the X-Y plane.

Different scenarios involving the placement of the eNodeBs were explored using this algorithm. The first scenario used random placement of the eNodeBs at varying angles from the actual UE and normally distributed random distances of 1.2 kilometers with a standard deviation of 400 meters. A Monte Carlo simulation of 100,000 iterations was performed for each increment in the number of eNodeBs, ranging from three to ten. In the same manner as the two-eNodeB simulation, the average distance from the approximation to the actual UE location for each increment of eNodeBs was recorded.

This completely random placement scheme of eNodeBs presented rare cases where a three eNodeB network produced a very inaccurate approximation estimate due to the third radius circle intersecting the other two circles at the same or near-same points. When this happens, the three-eNodeB network essentially devolved into a two-eNodeB network and was subjected to the approximation errors created with varying angles discussed previously. The case of a three-eNodeB network devolved to a two-eNodeB network with ambiguity due to unfortunate geometry, where eNodeB towers are represented with triangles, diamonds for the chosen points of intersection, a star for the estimated location, and a square indicating the actual UE location, are illustrated in Figure 17.

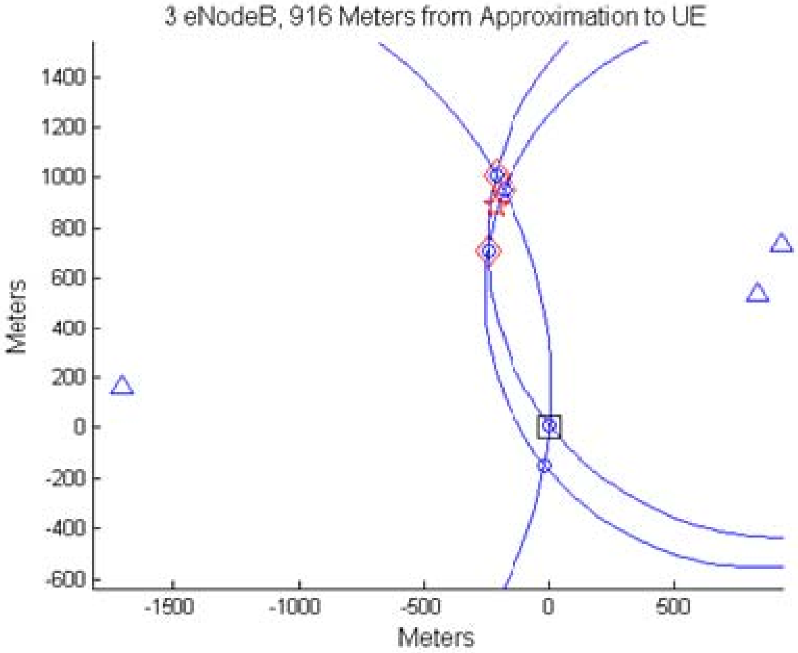


Figure 17. Inaccurate approximation situation with a 3 eNodeB network.

The next scenarios explored eNodeB placement with nonrandom distribution. Real world networks tend to follow some form of structure in their base station layout based upon maximum coverage with the fewest amounts of towers in support of subscriber density for different areas. The individual topography of these areas can prevent exact alignment to the structure, but towers are still generally spaced to support the area’s required demands. Two models were used to simulate structured eNodeB placement. First, the eNodeBs were placed with evenly spaced, fixed angles between them, based on the number of eNodeBs in the network, and the inter-site distances remained randomly distributed. The next model used the same even angle placement, but all eNodeBs at a fixed range of one kilometer from the UE.

All three scenarios were initially simulated in the Monte Carlo fashion of 100,000 iterations per each incremental number of eNodeBs, ranging from three to ten and using the radial distances generated from TA calculations without any standard deviation applied. As expected, the structured scenarios provided greater approximation accuracy than the randomly placed eNodeB scenario, as shown in the overall results displayed in Figure 18.

Returning to the issue of devolving three-eNodeB networks, we reiterate the importance of the fact that the approximation estimates are averages from all iterations of each particular geometry case. Some estimates that are hundreds of meters off are averaged along with very accurate results, which can be seen with the large location estimate error for a three-eNodeB network in Figure 18. Despite the overall average estimate between the approximation and the actual UE location as 68 meters, the simulation results in a greater number of accurate fixes than the unfortunate geometry cases like Figure 17. Sample plots from the multiple-eNodeB simulations, including an accurate three-eNodeB network that approximates the location of the UE within 5 meters, are contained in Appendix B.

Based on the results presented in Figure 18, the structured networks show error estimates less than 25 meters. The scenario with evenly spaced angles and fixed radial distances produced estimates with no significant error. An enhanced view of the error results is shown in Figure 19, and an interesting occurrence of greater degree of accuracy with even-numbered eNodeB networks is displayed. Expecting real-world networks to fall somewhere between structured and completely random geometries, we are safe in assuming UE location approximations to result on the average to less than 40 meters. Sample plots generated from the different multiple eNodeB simulations, with estimate errors well within the 40 meters average, are presented in Appendix B.

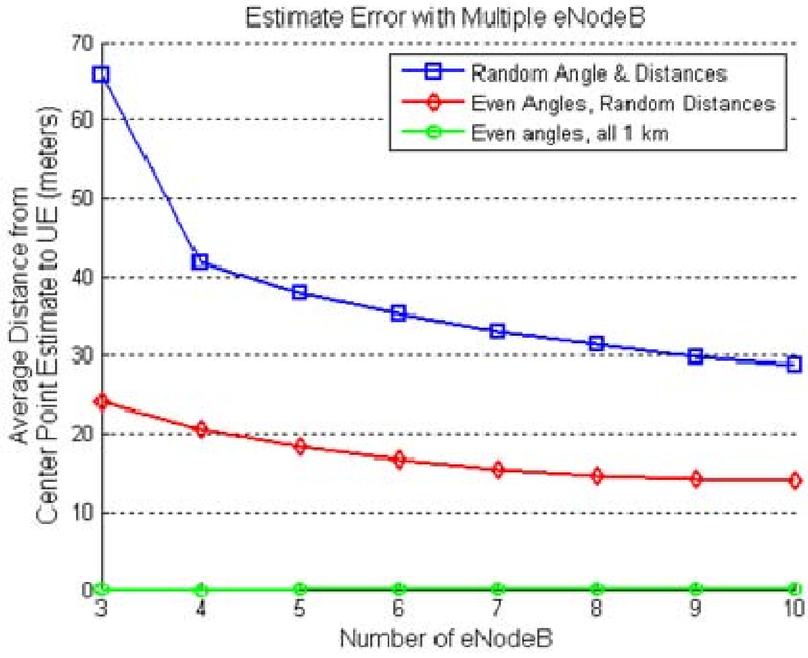


Figure 18. Average Distance from Estimate to UE with Multiple eNodeBs.

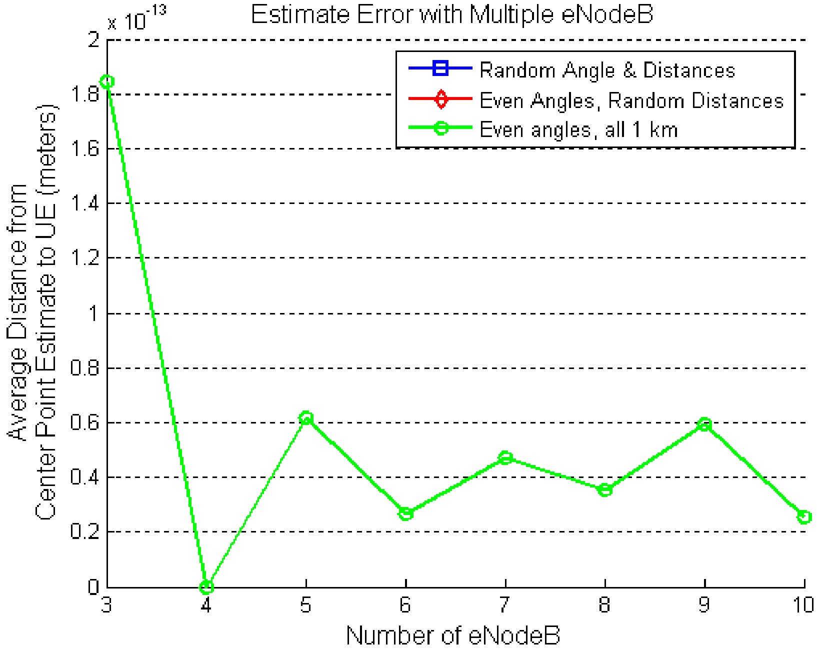


Figure 19. Error analysis of multiple eNodeBs with even angles and fixed radial distances.

A second simulation of the three scenarios was performed, this time with the randomly selected TA measurement standard deviation of 0.5 meters, quantization error of 78.125 meters, and bias of 39.0625 meters. This increase in radial distances resulted in larger approximation errors across the board, as expected. The results, which follows the same distribution in the previous results, are displayed in Figure 20. The scenario with evenly-spaced eNodeBs at fixed distances produces errors that are not nearly as accurate as the simulation without TA standard deviation. Regardless, the approximation error falls within 40 meters for the most ideal network geometry. Again taking into account that real-world network geometries fall between ideal and random tower placement, as well as the high likelihood that TA variations will exist, location estimate errors fall on the average to less than 90 meters away from the actual UE location.

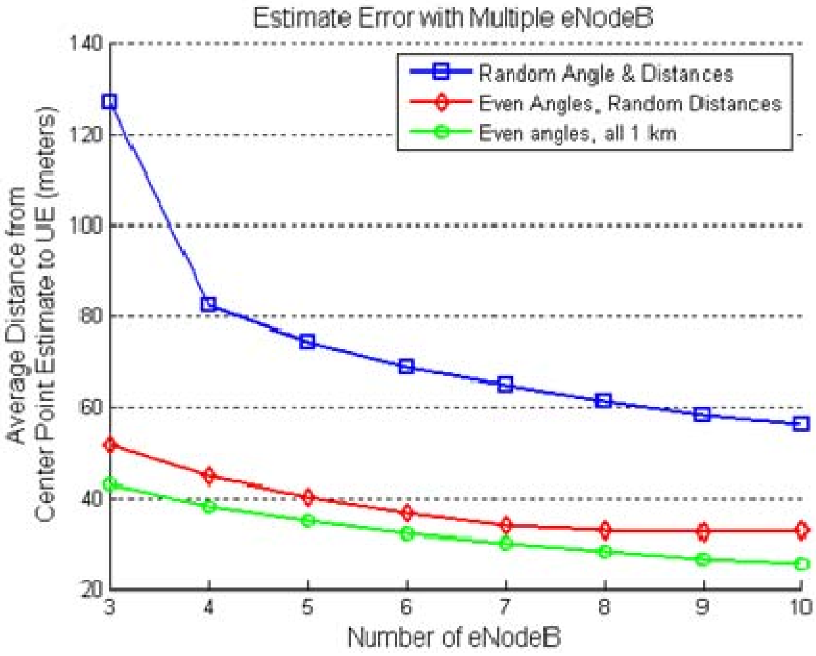


Figure 20. Average distance from estimate to UE with multiple eNodeBs, standard deviation applied.

For further analysis of the collected data, the same scenarios can be compared with circular error probability (CEP), where the radius within which 50 percent of the samples lie is assessed rather than using the average total distance from the approximation to the actual UE location. Essentially, the median value of the 100,000 iterations is used versus the mean average of all iteration results. The CEP for the same scenarios as before is shown in Figure 21. The CEP results for the same scenarios are displayed in Figure 22, but standard deviation applied to TA and radii is included. The same distribution trends based on the random and ideal scenarios are seen when analyzing the results, but the circle containing 50 percent of the estimates has a smaller radius than the average distance to the estimates calculated.

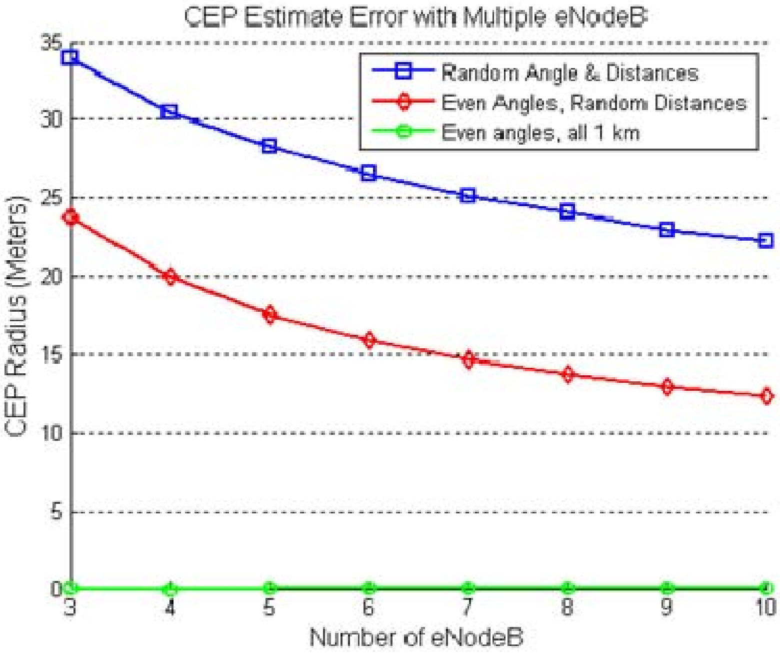


Figure 21. Circular Error Probable from multiple eNodeB simulations.

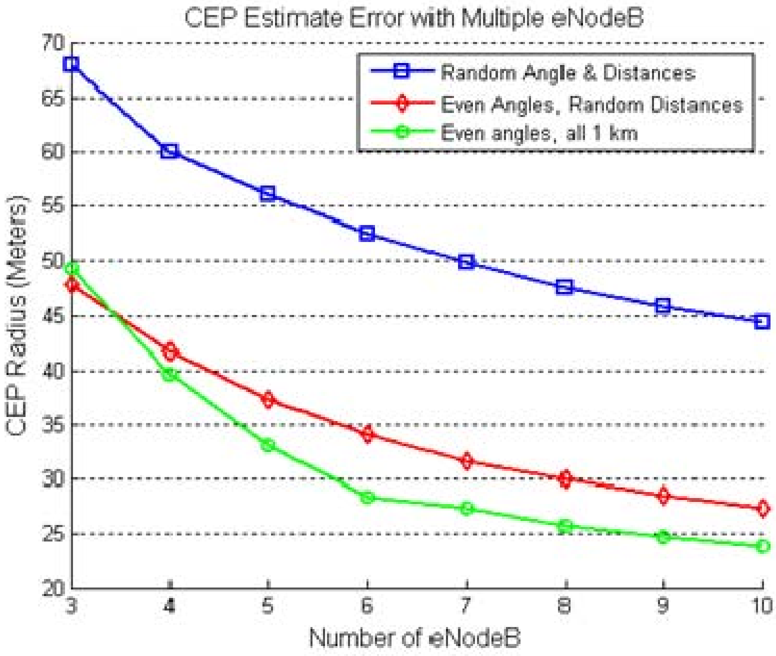


Figure 22. Circular Error Probable from multiple eNodeB simulations, standard deviation applied.

Comparing the CEP results based upon the presence or lack of standard deviation and assuming that accurate eNodeB locations are known, we are safe in assuming that in more than 50 percent of all cases, a UE can be accurately located within 60 meters using a two-dimensional mapping scheme.

Calculations derived from LTE specifications to computer-simulated LTE variable networks portrayed within a two-dimensional mapping were applied in this chapter. With Monte Carlo simulation schemes, computer models revealed that when using average distance from an approximated position to actual UE location or CEP as a measure, an LTE UE can invariably be located within the distance per unit of TA, less than 60 meters from the true position. Excellent location capability, suitable for mobile location requirements delineated by organizational entities discussed in Chapter I, were shown by the simulation results. Geolocation of an LTE UE using a three-dimensional model, discussion of the relationship of TA generated radii between a UE and eNodeB with a third-coordinate system, and application of these aspects using computer simulation are investigated in the next chapter.

IV. THREE-DIMENSIONAL SIMULATIONS

A. METHOD OF APPROXIMATIONS

When locating a point on the earth, one generally thinks in two dimensions. Longitude and latitude, the military grid reference system (MGRS) and other map datum depict a location in terms of two variables. However, the truth of the matter is the eNodeB and UE are calculating a TA-based radius with a three-dimensional distance, which proves longer than the two-dimensional distance. As the elevation, or Z-axis, between a UE and eNodeB varies so does the error [18]. This is best illustrated by a simple triangle, as shown in Figure 23 (After [18]).

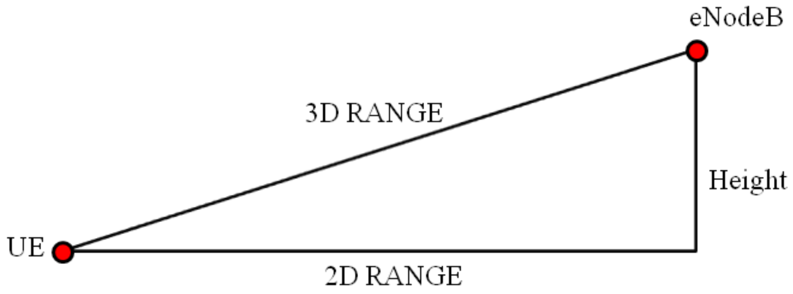


Figure 23. Illustration of difference between 2D and 3D ranges (After [18]).

In no way does the addition of the Z-axis variable for height suggest the two-dimensional mapping with known TA investigated in Chapter III was not a viable solution to geolocating an LTE UE. On the contrary, the method of approximation for a two-dimensional solution using only the X-Y plane poses little difference to approximating a three-dimensional solution on the X-Y-Z plane. The same method of trilateration can be employed. However, where the two-dimensional simulation method used the geometry of triangles for determining relative location approximations, three-dimensional trilateration requires the geometry of spheres. This technique is employed by GPS, where satellites in space essentially act as the center points for spheres with a radius equal to the distance between the satellite and the GPS device requesting location updates. The estimated point of intersection of three or more satellite spheres represents the likely location of the GPS device, just as the intersection points of circles can estimate a probable location. For the purposes of geolocating an LTE UE, the height of an eNodeB is accounted for on the Z-axis, where the top of the tower represents the center point of a sphere, with the TA-based distance between the UE and the eNodeB as the radius.

The three-dimensional approach to simulating geolocation of an LTE UE uses the same parameters established for the two-dimensional simulations. The simulation employs a flat Earth model with calculations derived in Cartesian coordinates. Radii from the eNodeB to the UE continue to assume free space propagation at the speed of light and each unit of TA increases the range radius from the eNodeB by 78.125 meters. Best case scenarios assume calculations of range radii based on an absolute TA amount of 78.125 meters, and secondary runs of the simulations offset the best case scenario by introducing a randomly selected measurement standard deviation of 0.5 meters to the TA values, 78.125 meter quantization error and 39.0625 meter bias to the radii.

B. LIKELY LOCATION CALCULATIONS

On the two-dimensional plane, the three situations with radius rings that must be accounted for included two radius circles separated without touching, one radius circle completely contained in another, and radius circles that intersect. The same situations apply when using two spheres. However, when two spheres intersect, the addition of the third plane introduces a vast number of shared intersection points in contrast to two circles only having two points of intersection. Therefore, the addition of a third sphere is necessary to calculate a shared intersection point that correlates to the actual UE position for all situations.

Using the Pythagorean Theorem and the law of cosines as before, we found the spheres from the eNodeBs using the TA-based radii, the X-Y plane coordinates of each eNodeB, and the height of the eNodeB on the Z-plane. This results in a system of equations that can be used with linear algebra to determine the most common point of intersection corresponding to the actual UE location.

To elaborate, in terms of Cartesian coordinates, four spheres equate to a system of four equations:

$$\begin{aligned}
\left(x-x_1\right)^2+\left(y-y_1\right)^2+\left(z-z_1\right)^2&=R_1{}^2\\
\left(x-x_2\right)^2+\left(y-y_2\right)^2+\left(z-z_2\right)^2&=R_2{}^2\\
\left(x-x_3\right)^2+\left(y-y_3\right)^2+\left(z-z_3\right)^2&=R_3{}^2\\
(x-x_4)^2+(y-y_4)^2+(z-z_4)^2&=R_4{}^2.
\end{aligned} \tag{9}$$

Points$\left(x_1,y_1,z_1\right)$,$\left(x_2,y_2,z_2\right)$, $\left(x_3,y_3,z_3\right)$, and (x_4,y_4,z_4) correspond to the center points of the four spheres calculated with the law of cosines and their associated radii R_1, R_2, R_3,and R_4, respectively. After substituting values for center point coordinates and radii, we further reduced the system by subtracting the equation for the first sphere from the equations of the remaining three spheres. The reduced system can now be expressed as:

$$\begin{aligned}
(x_2-x_1)^2x+(y_2-y_1)^2y+\left(z_2-z_1\right)^2z&=d_1\\
(x_3-x_1)^2x+(y_3-y_1)^2y+\left(z_3-z_1\right)^2z&=d_2\\
(x_4-x_1)^2x+(y_4-y_1)^2y+\left(z_4-z_1\right)^2z&=d_3
\end{aligned} \tag{10}$$

where

$$\begin{aligned}
d_1&=R_1{}^2-R_2{}^2-x_1{}^2+x_2{}^2-y_1{}^2+y_2{}^2-z_1{}^2+z_2{}^2\\
d_2&=R_1{}^2-R_3{}^2-x_1{}^2+x_3{}^2-y_1{}^2+y_3{}^2-z_1{}^2+z_3{}^2\\
d_3&=R_1{}^2-R_4{}^2-x_1{}^2+x_4{}^2-y_1{}^2+y_4{}^2-z_1{}^2+z_4{}^2.
\end{aligned} \tag{11}$$

To find the common point of intersection from (10) and (11), the use of matrices in the form of $\mathbf{Ax}=\mathbf{b}$ can be employed, where matrix **A** encompasses the coefficients from (10), vector **b** holds the constants from (11), and vector **x** contains the unknown coordinates for the intersection(x,y,z). This is visualized as

$$\begin{pmatrix} (x_2-x_1)^2 & (y_2-y_1)^2 & (z_2-z_1)^2 \\ (x_3-x_1)^2 & (y_3-y_1)^2 & (z_3-z_1)^2 \\ (x_4-x_1)^2 & (y_4-y_1)^2 & (z_4-z_1)^2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix}. \tag{12}$$

With (12), the unknown point of intersection in vector **x** is simply the inverse of the coefficient matrix **A** multiplied by the constant vector **b**, $\mathbf{x = A^{-1}b}$.

As efficient and simple as this likely location calculation may be, there still remains the potential for the calculations to go awry. Primarily, an inverse of the coefficient matrix **A** only exists when **A** is a square $n \times n$ matrix. This condition can only be met in the case of a four eNodeB network, where the system of equations reduces to three equations for three unknowns. Other network sizes produce an under-determined or over-determined system of equations and do not have a unique solution. Second, eNodeBs with fixed distances and tower heights spread at evenly spaced angles across the coordinate plane can cause some of the coefficients in matrix **A** to equate to zero. For fixed towers, the Z-coordinates always cancel out, and X-Y coefficients may cancel out as well, dependent upon their calculated values based on the law of cosines and TA radii. In essence, matrix **A** becomes a singular matrix, where the determinant of the matrix is zero and has no inverse, and $\mathbf{Ax = b}$ does not exist or is not unique. These problems are frequently encountered with GPS technology and can be mitigated through the use of *pseudo-ranging*.

In linear algebra, a resolution to the first problem, where **A** is not a square matrix, is to not rely on **A** having an inverse and use Gaussian elimination with partial pivoting to find a least-squares, or "best fit" solution. This employs the theorem that if **A** is an $m \times n$ matrix of rank n, the normal equations $\mathbf{A^T Ax = A^T b}$ have a unique solution $\hat{\mathbf{x}} \mathbf{= (A^T A)^{-1} A^T b}$, and $\hat{\mathbf{x}}$ is the unique least-squares solution to the system $\mathbf{Ax = b}$. If **A** is a square matrix, this theorem will provide the same solution as $\mathbf{x = A^{-1}b}$.

While suitable for the problem of nonsquare coefficient matrices, the solution presented above does not satisfy the second problem presented, where X-Y-Z coefficients cancel to zero, and **A** becomes a singular matrix. Essentially, the rank, or number of pivots, of **A** for an $m \times n$ matrix becomes k, where $k < n$. In these cases, the Moore-Penrose pseudo-inverse can be used to compute a least-squares solution to a system of equations. The $m \times n$ matrix $\mathbf{A}^{+}$ is the pseudoinverse of a matrix $\mathbf{A}$ if $\mathbf{A}^{+}$ satisfies four Moore-Penrose conditions: $\mathbf{AA}^{+}\mathbf{A} = \mathbf{A}$, $\mathbf{A}^{+}\mathbf{AA}^{+} = \mathbf{A}^{+}$, $\mathbf{AA}^{+}$ is a symmetric matrix, and $\mathbf{A}^{+}\mathbf{A}$ is a symmetric matrix.

Singular value decomposition (SVD) is used to calculate $\mathbf{A}^{+}$ by replacing every nonzero entry in matrix **A** by its reciprocal and then taking the transpose of the resulting matrix. For a linear system of spheres $\mathbf{Ax} = \mathbf{b}$, the least-squares solution with the smallest norm $\|\mathbf{x}\|$ becomes $\|\mathbf{x}\| = \mathbf{A}^{+}\mathbf{b}$, giving a "best fit" point of intersection for all spheres in the system and a likely location approximation.

To demonstrate the potential for geolocating a UE within a real world LTE network in three dimensions, software-implemented simulations were again produced using MATLAB. Multiple eNodeB scenarios were simulated using the mean distance per unit of TA of 78.125 meters derived from calculations in Chapter II. The same scenarios were then executed a second time with the application of a randomly selected measurement standard deviation of 0.5 meters per unit of TA, 78.125 meter quantization error standard deviation and 39.0625 meter bias to the radii and compared against the first run.

C. THREE BASE STATION SIMULATION

The first three-dimensional simulation uses the algorithm for solving a system of equations for spheres presented with three eNodeB radii to approximate the location of a UE. Similar to the two-dimensional simulations in Chapter III, the UE's true location is the origin on the X-Y-Z plane. The simulation creates three eNodeBs at varying degrees in relation to the UE, each with normally distributed random distance with a mean of 1.2 kilometers from the UE, standard deviation within a range of 400 meters, and TA per the calculated value of 78.125 meters. Since the value of TA is discrete, the same process of rounding the radial distance to a whole unit of TA is used. Tower height of the eNodeBs were generated with normally distributed random heights having a mean of 305 meters from the ground, with a standard deviation of 250 meters to facilitate a minimum eNodeB height of 55 meters.

Ten trials were conducted with a Monte Carlo simulation of one hundred thousand iterations per trial. The average distance from the approximation point to the actual UE location for each separate trial was recorded. The simulation was then repeated with the addition of measurement standard deviation of 0.5 meters per unit of TA, 78.125 meter quantization error and 39.0625 meter bias to introduce fluctuations in the radii of the spheres and compared with the first simulation results.

In both cases, the simulations provided extremely accurate results and were far better than all two-dimensional simulations. Another interesting find with the three eNodeB network simulation was that the introduction of standard deviation to TA and quantization error to the radii had little effect on the approximation error, and all results were consistently less than one meter of distance error between the approximation and the true UE location. The results of the simulation for all ten trials are shown in Figure 24, and shows the average distance between the estimated position and the actual location of the UE was less than 10 centimeters. A Mean Radial Spherical Error (MRSE) analysis was also conducted, the results are shown in Figure 25. To calculate MRSE, the variances for the X, Y, and Z coordinates of all 100,000 Monte Carlo estimates are summed, and the square root of this value is taken:

$$MRSE = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2}. \tag{13}$$

The radial value of MRSE, centered on the origin and actual UE location, shows where 61% of the location estimates are contained. Based on these results, it is safe to assume a UE can be accurately located in a three eNodeB network within a 20 meter radius. Sample plots generated from the three-dimensional three eNodeB simulations, with estimate errors well within the 10 centimeter average, are presented in Appendix C.

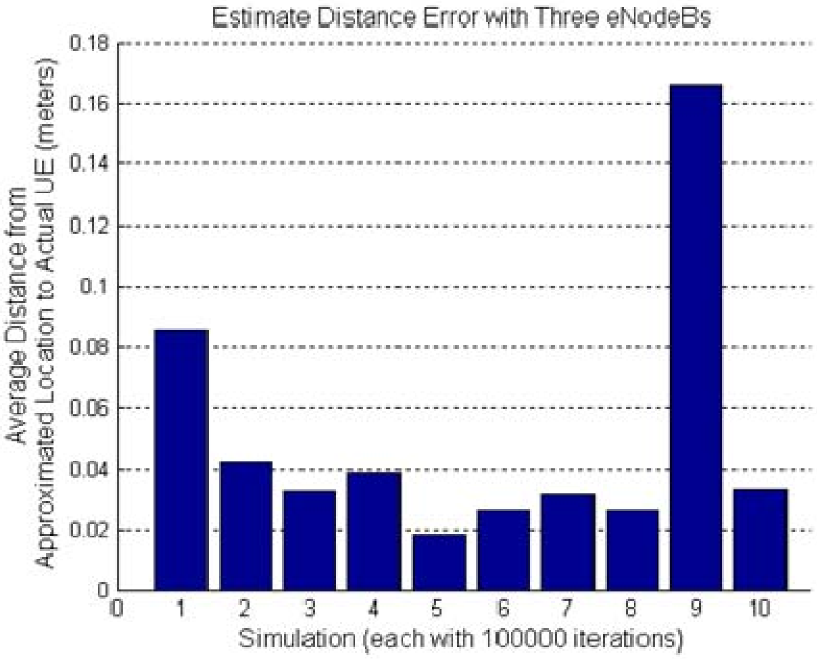


Figure 24. Three-dimensional average distance from estimate to UE with 3 eNodeBs.

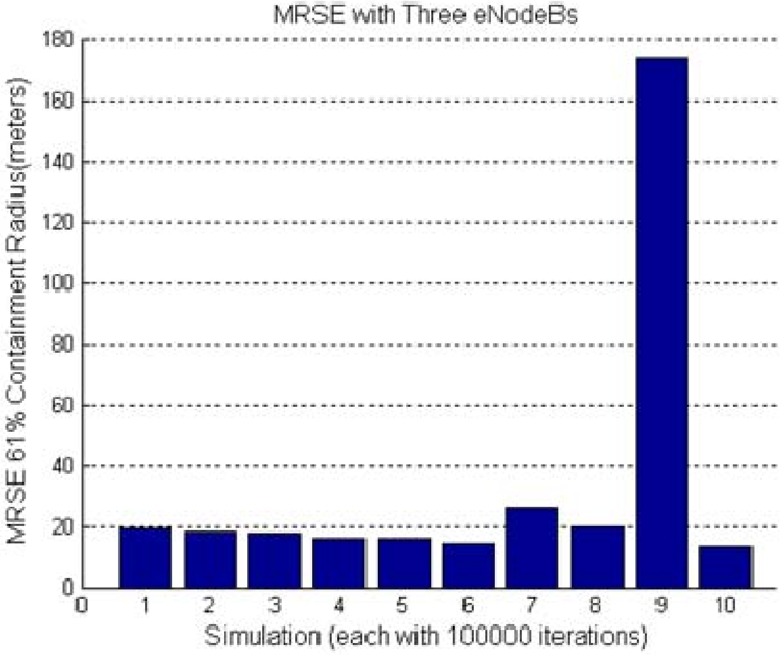


Figure 25. Mean Radial Spherical Error from 3 eNodeBs simulation.

D. FOUR BASE STATION SIMULATIONS

The next simulation investigates a four eNodeB LTE network. The same parameters were used as the previous simulation with eNodeB placement at random angles from the UE, mean distances of 1.2 kilometers with standard deviation of 400 meters, radii based on the calculated TA value of 78.125 meters, and random tower heights of 305 meters from the ground with standard deviation of 250 meters.

Ten trials with a Monte Carlo simulation scheme of 100,000 iterations per trial were executed, and the average distance between the approximations and the actual UE location were recorded. The ten trials were then repeated with eNodeB placement at evenly spaced angles while maintaining random heights and distances. Next, ten trials were run with eNodeB placement at evenly spaced angles and random tower heights but fixed distances of 1732 meters. Finally, a fourth set of ten trials were conducted for eNodeB placement at evenly spaced angles, fixed distances of 1732 meters, and fixed tower heights of 305 meters each, portraying the most ideal network setting. The algorithm for a system of equations discussed previously was used to calculate the approximation coordinates, to include employment of finding a least-squares solution to mitigate possible singular matrices when running the ideal network scenario test.

With the addition of a fourth eNodeB-generated sphere, approximation accuracy on the average consistently improved in comparison to the two-dimensional simulations, but diminished in comparison to the previous three eNodeB simulation. In retrospect, a similar situation occurs with a three-dimensional four eNodeB network just as with a two-dimensional three eNodeB network, where very large approximation errors are weighed in with several accurate position estimates.

An example of a four eNodeB network with a gross approximation error is displayed in Figure 26, where the distance between the approximation and the UE was 134 meters. The root cause of the large fluctuation in accuracy can be centralized within the system of equations for a four eNodeB network. Having four spheres equating to four equations and then reducing them to three equations, we see that the algorithm is left with three equations to solve for three unknowns. The mathematics becomes ambiguous when dealing with square terms, essentially resulting at times to a quadratic solution with two possible values. In all cases, the largest variation can be found in the approximated height value for Z, which can be a very large positive or negative number, or essentially very high above or very far below the surface of the Earth.

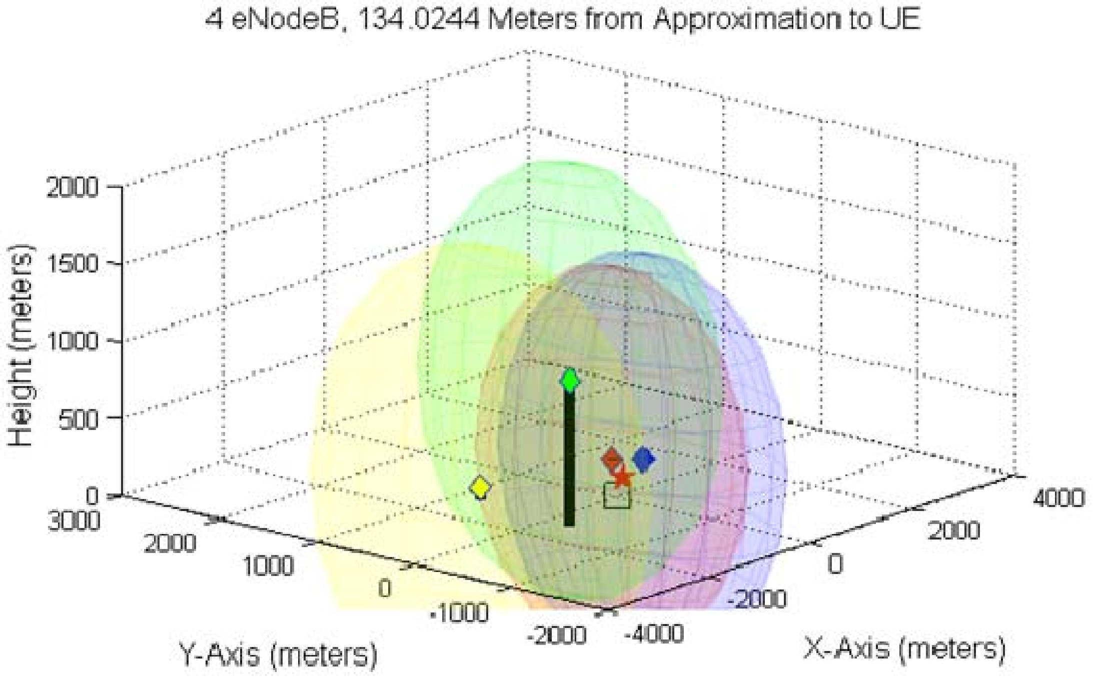


Figure 26. Example of gross three-dimensional approximation error with a 4 eNodeB network.

Regardless of the possibility for large approximation errors, the simulations resulted in accuracy far better than the two-dimensional simulations. The results of the four scenarios involving eNodeB placement and height for all ten trials of the simulation are displayed in Figure 27. Despite changes in the network due to random eNodeB placement and characteristics, the average distance between the approximation and the actual UE location fell within 5 meters. The most ideal scenario of evenly spaced eNodeBs with fixed distances and tower heights had exactly zero error. However, the results of the MRSE analysis shown in Figure 28 shows an extremely high degree of variance within the Monte Carlo simulation results, and suggests that, on the average, 61% of the location estimates can be contained within a radius of 1000 meters from the actual UE.

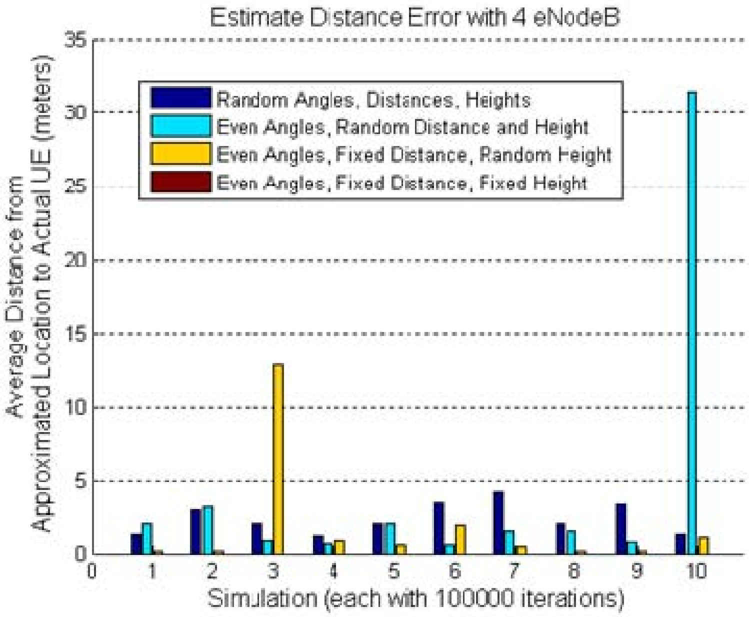


Figure 27. Three-dimensional estimate error analysis with 4 eNodeBs.

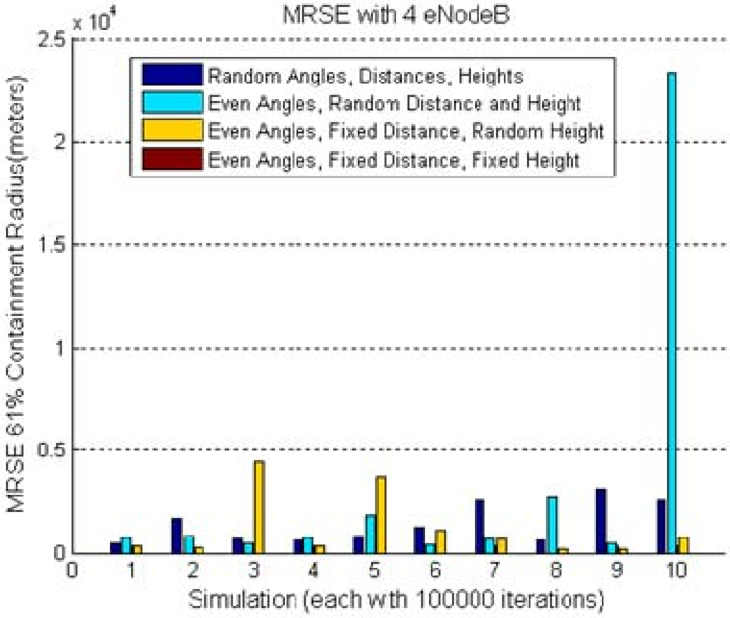


Figure 28. Three-dimensional MRSE error analysis with 4 eNodeBs.

The same simulations were performed a second time to evaluate the effects of introducing standard deviation to the TA and radii of the four spheres. With standard deviation applied, the average distance between the approximation and the actual UE location increased slightly for the most part, including a small but noticeable increase of estimation error in the ideal network scenario. As indicated in Figure 29, the error generally fell somewhere less than 10 meters.

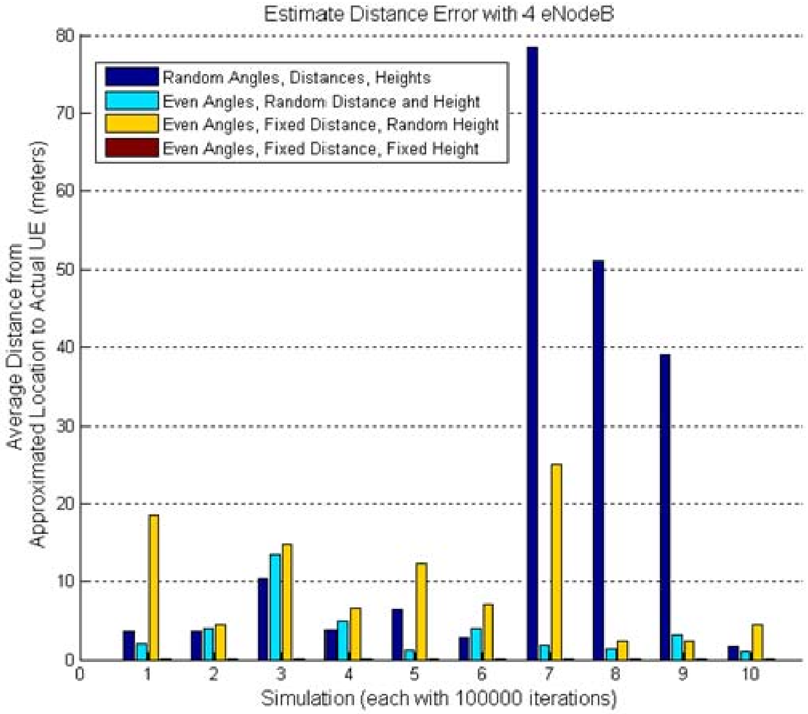


Figure 29. Three-dimensional estimate error analysis with 4 eNodeBs, standard deviation applied

The MRSE results for the same simulation were calculated and are shown in Figure 30. Similar to the results in the previous simulation, a degree of high accuracy with low precision is seen with the four eNodeB network. By introducing standard deviation, the four eNodeB network shows 61% containment within an average radius of 10,000 meters. Sample plots generated from the three-dimensional four eNodeB simulations are presented in Appendix C, including examples where the error was less than one meter.

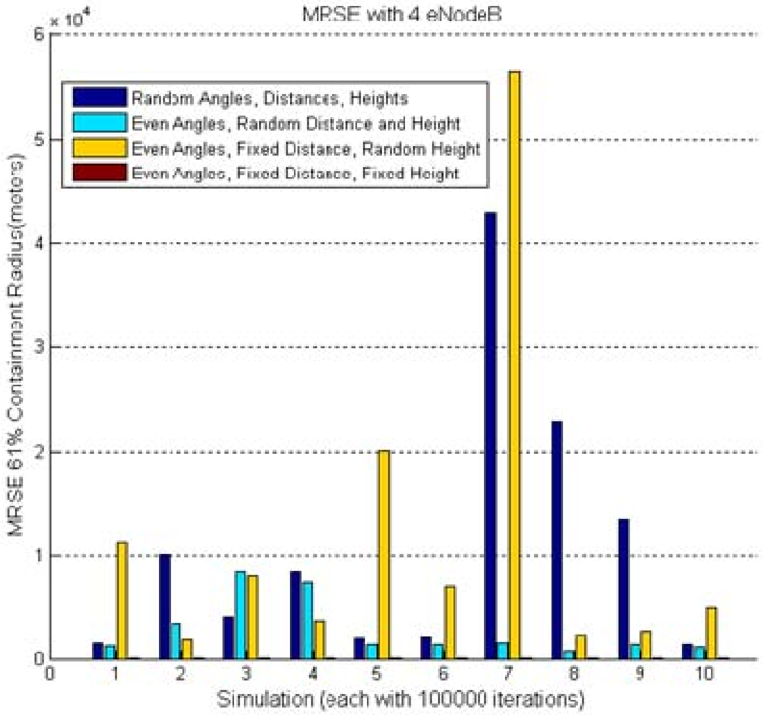


Figure 30. Three-dimensional MRSE error analysis with 4 eNodeBs, standard deviation applied.

As stated previously, the gross approximation error derived primarily from the calculated Z-coordinate. In all cases of gross approximation results, the X-Y coordinates remained sound, correlating very closely to the true X-Y coordinates of the actual UE. In a real-world situation, one could assume that a UE would not be located at large distances above or below the surface of the Earth. This does not suggest that a height coordinate for a UE location estimate should be ignored, but, for purposes of geolocating rapidly, two accurate X-Y coordinates more than suffice to approximate a UE location.

To justify this argument, an example of a four eNodeB network with gross approximation error is presented in Figure 31. In this example, the algorithm calculated the Z-coordinate of the approximation as -1540 meters, or better said 1540 meters below the surface of the Earth. However, when using Pythagorean Theorem to calculate the distance from the approximation to the actual UE location using only X-Y coordinates in the equation, we can locate the UE position accurately, within 85 centimeters.

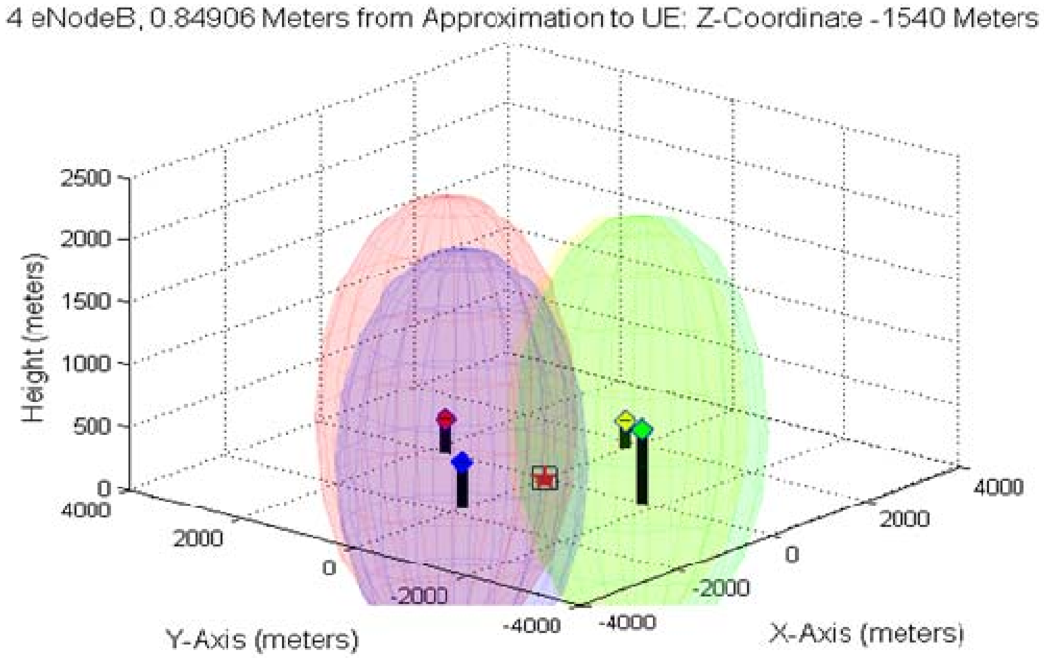


Figure 31. Example of an accurate location estimate with a 4 eNodeB network using approximation X-Y coordinates and disregarding the gross Z-coordinate value.

E. MULTIPLE BASE STATION SIMULATIONS

The final three-dimensional simulation tests approximation accuracy used multiple eNodeBs and followed the same approach as the two-dimensional multiple eNodeB simulation. First, eNodeBs were placed at random angles in relation to the UE at the origin. Distances of the eNodeBs followed a normally random distribution with a mean of 1.2 kilometers and standard deviation of 400 meters. Tower heights of the eNodeBs were randomly generated with a mean of 305 meters and standard deviation of 250 meters to facilitate a minimum tower height of 55 meters. TA-based radii of the spheres generated between the eNodeBs, and the UE used the calculated value of 78.125 meters per unit of TA. The simulation included the three and four eNodeB networks previously simulated and continued to test increments in the number of eNodeBs, up to a network with ten eNodeBs. A Monte Carlo simulation of 100,000 iterations was conducted for each network size, and the average distances from the approximation to the actual UE location was recorded.

Next, the simulation was repeated with eNodeB placement at evenly spaced angles in relation to the actual UE location at the origin and maintained random distances and tower heights. A third simulation was executed leaving evenly spaced eNodeB placement and random tower heights, but changed the distances of the eNodeBs to a fixed value of 1732 meters. Finally, a fourth simulation tested the most ideal network scenario, using evenly spaced angles for eNodeB placement, fixed distances of 1732 meters, and fixed tower heights of 305 meters. The algorithm for a system of equations of spheres and the use of matrices to solve for the approximation coordinates was employed, including the use of a least-squares solution for cases involving singular coefficient matrices with the most ideal network scenario. For all scenarios, the simulations included factorization of the Z-coordinate with the four eNodeB network.

The results of the simulation, displayed in Figure 32, showed best accuracy obtained with a network encompassing only three eNodeBs. The largest approximation error occurred in the four eNodeB network, which was expected to have a higher degree of error than the network using only three eNodeBs. Interestingly, the margin of approximation error with five or more eNodeBs remained consistent, despite the increment of additional towers, and well below the error seen in the four eNodeB trial. Another point of interest is the smaller degree of approximation error in the three eNodeB network when compared to five or more eNodeBs. This suggests that, with a system of equations to solve for three unknowns, a system of four or more equations would not be as precise as a system of only two equations. In the case of the ideal network scenario, the results for all network sizes approximated the exact location of the UE with zero error. The average estimated location results are displayed in Figure 32 and an LTE UE location can be approximated well within 1 meter, based upon a lack of standard deviation.

The MRSE for the approximation error is shown in Figure 33, but the results are ambiguous due to the 61% radial containment for the four eNodeB network ranging up to 2200 meters. The MRSE results are enhanced in Figure 34, and the higher accuracy and precision with a three eNodeB network is validated. The five eNodeB network indicates

61% estimate containment within 12 meters. Interestingly, the precision increases by incrementing the number of eNodeBs past five and gives an average of 61% containment within 3 meters.

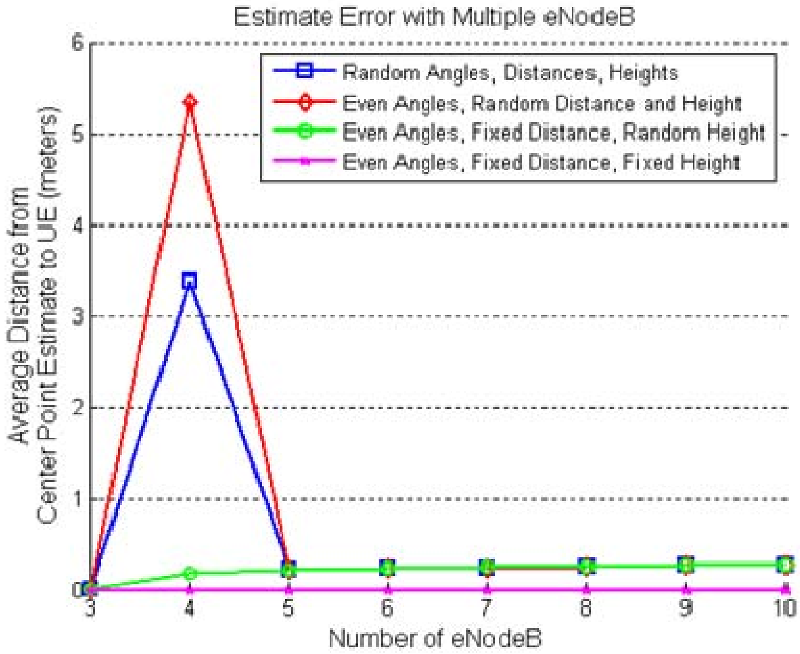


Figure 32. Three-dimensional estimate error analysis with multiple eNodeBs.

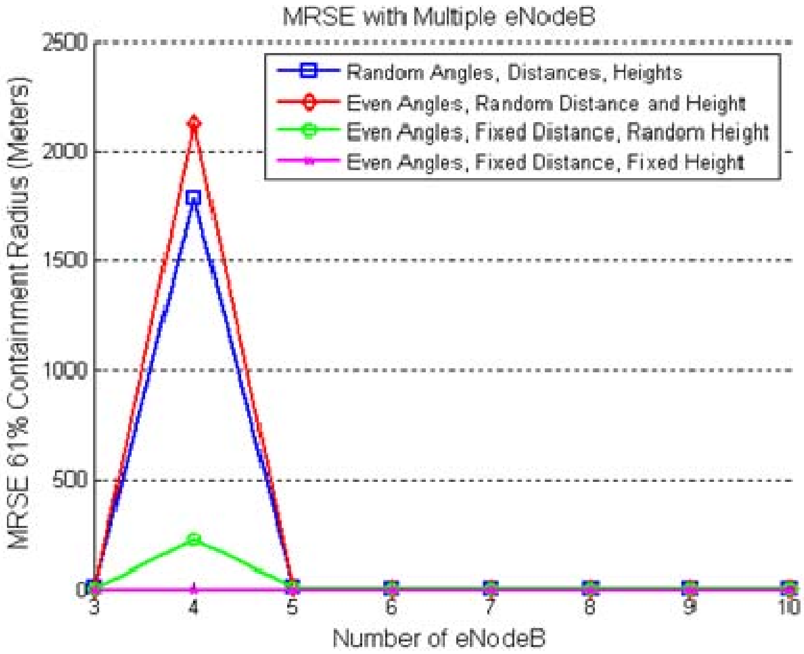


Figure 33. Three-dimensional MRSE error analysis with multiple eNodeBs.

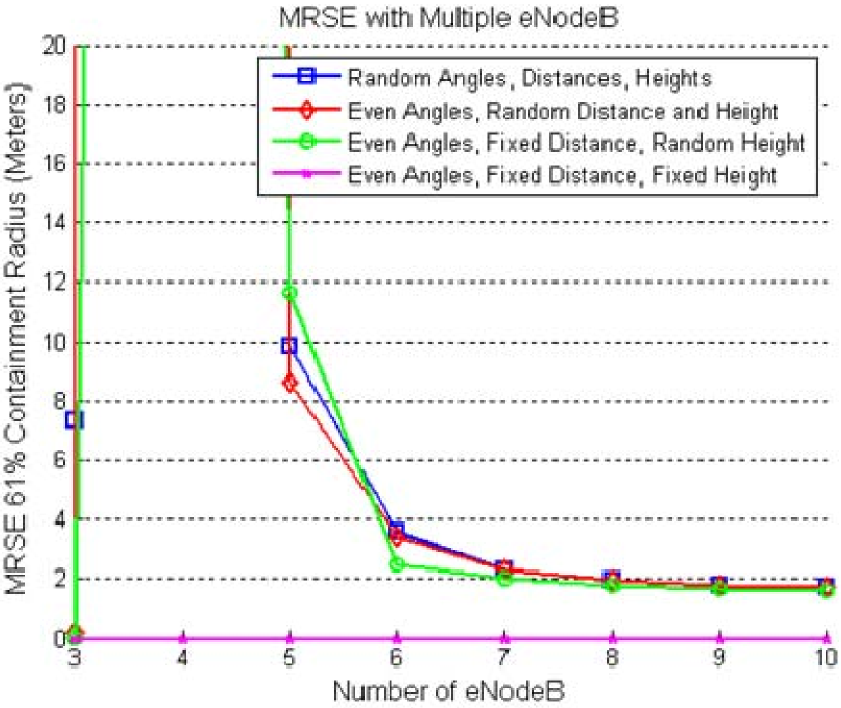


Figure 34. Three-dimensional MRSE error analysis with multiple eNodeBs (enhanced view).

Next, the same simulations were executed to include the randomly selected measurement standard deviation of 0.5 meters per unit of TA, 78.125 meter quantization error and 39.0625 meter bias to induce fluctuation in the radii of the eNodeB spheres. The results of this simulation are shown in Figure 35, showing high accuracy with average estimate results falling within 50 centimeters. The three eNodeB network again displayed the highest accuracy, a four eNodeB network susceptible to ranging error due to ambiguous results and less than 1 meter accuracy for networks of five or more eNodeBs.

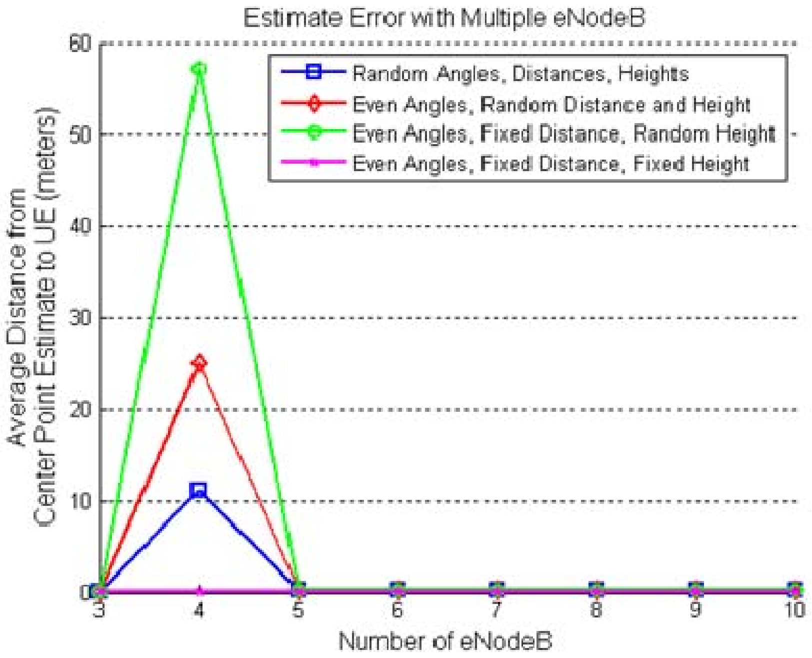


Figure 35. Three-dimensional estimate error analysis with multiple eNodeBs, standard deviation applied.

The MRSE analysis for the same simulation, shown in Figure 36, indicated extremely high variance in the four eNodeB network. A closer look at the MRSE results is displayed in Figure 37. With standard deviation applied, a three eNodeB network had 61% containment of location estimates falling within a radius of less than 1 meter. The five eNodeB network had 61% containment of location estimates within 35 meters, and network sizes from six to ten eNodeBs had 61% containment well within 10 meters. Also, while not accurately portrayed in these figures, the addition of standard deviation did cause approximation error in the ideal network scenario for all network sizes but was infinitely very small and negligible.

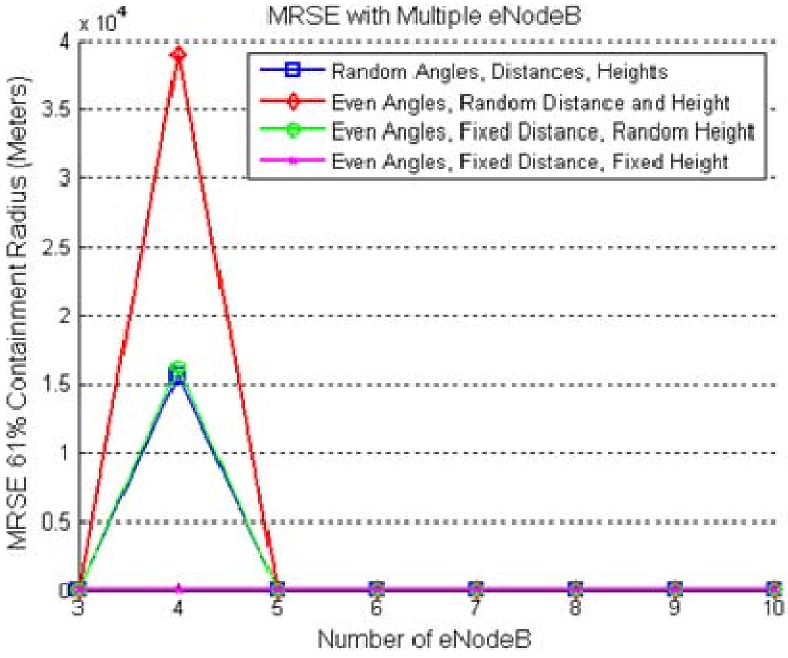


Figure 36. Three-dimensional MRSE error analysis with multiple eNodeBs, standard deviation applied.

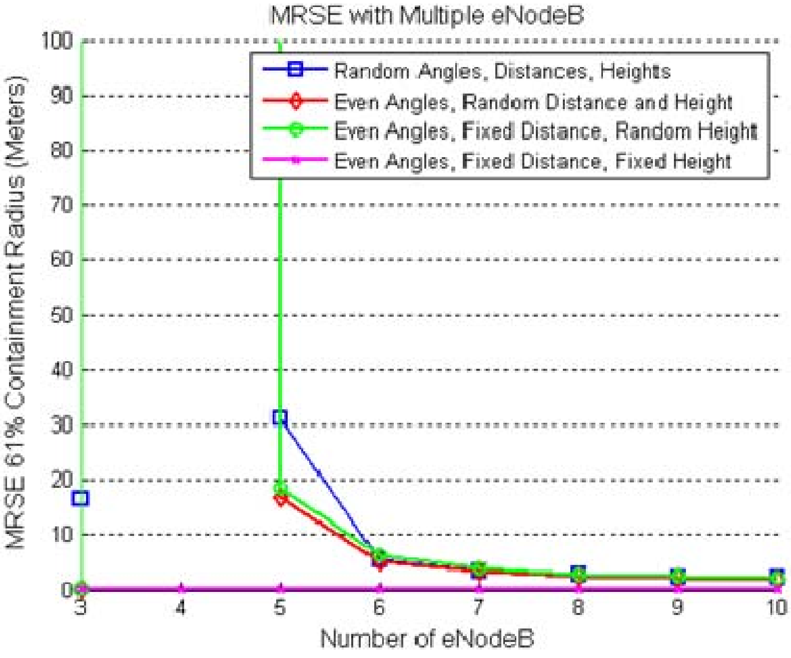


Figure 37. Three-dimensional MRSE error analysis with multiple eNodeBs, standard deviation applied (enhanced view).

A final simulation was conducted with the same parameters as the previous run, maintaining the measurement standard deviation of 0.5 meters per unit of TA, 78.125 meter quantization error and 39.0625 meter bias. For the four eNodeB network, the Z-coordinate for estimated UE height was omitted from the approximation calculation, relying solely on the derived X-Y coordinates. The results of the average distance between the approximation and actual UE location are shown in Figure 38. As before, the approximation error for the most ideal network scenario remained negligible, and the three eNodeB network maintained the most accurate UE position estimates.

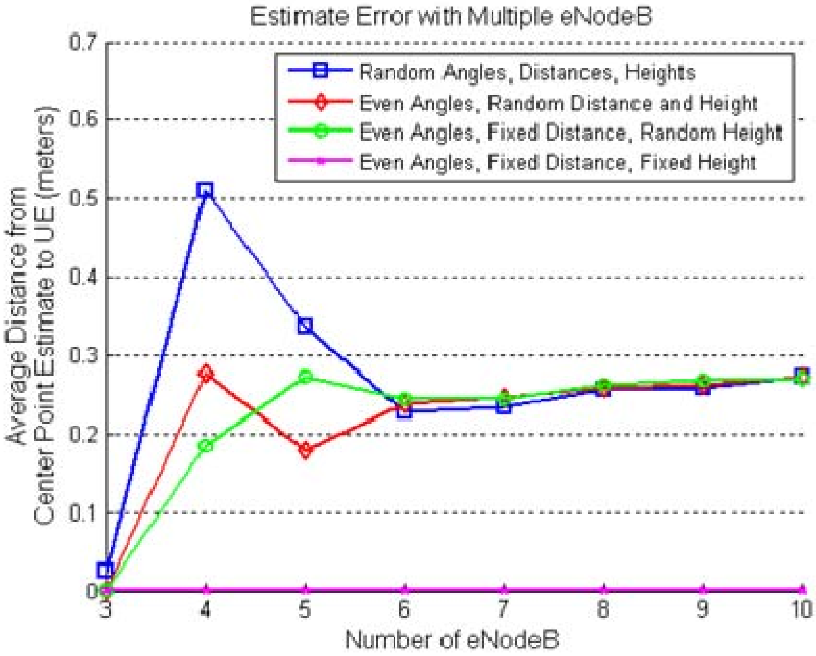


Figure 38. Three-dimensional estimate error analysis with multiple eNodeBs, standard deviation applied and disregarding approximation Z-coordinate in the four eNodeB network.

By omitting the Z-coordinate from the approximation calculation, the four eNodeB network displayed approximation accuracy well within 0.5 meters for the totally random eNodeB placement and height scenario. The results also show a trend with structured networks, where incrementing the number of eNodeBs used in the network causes an increase in error up to 30 centimeters when geolocating a UE.

The MRSE analysis results shown in Figure 39 give indication of high variance in the four eNodeB network for the worst case scenario, despite the omission of the Z-coordinates. However, the four eNodeB network had a significant increase in precision with the other scenarios involving a more ideal eNodeB placement scheme.

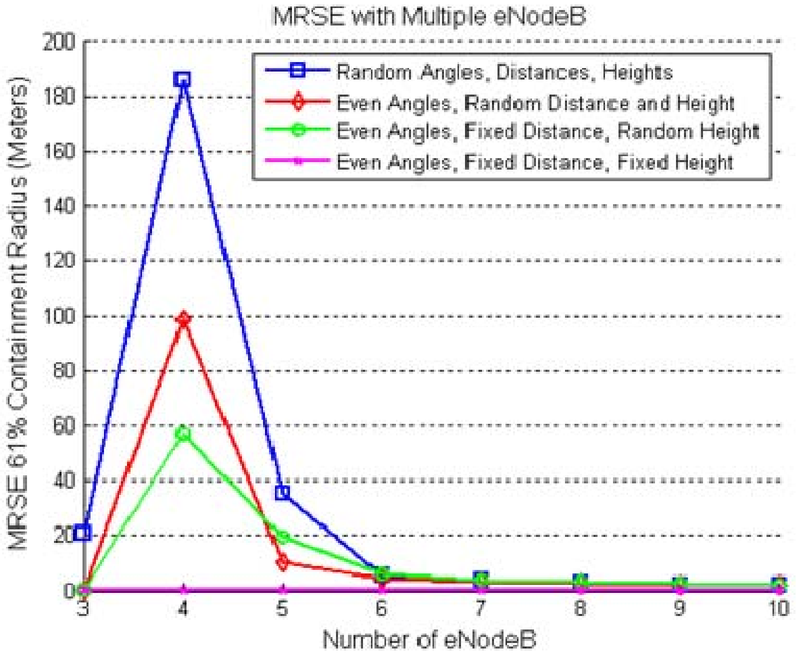


Figure 39. Three-dimensional MRSE error analysis with multiple eNodeBs, standard deviation applied and disregarding approximation Z-coordinate in the four eNodeB network.

To better analyze the MRSE for all network sizes, an enhanced view of the results with standard deviation and omission of the Z-coordinate in the four eNodeB network is shown in Figure 40. Again, the three eNodeB had 61% containment of estimate results falling within a radius of less than 1 meter, the four eNodeB network within a radius of 100 meters, a five eNodeB network within 35 meters, and all other network sizes well within 10 meters.

Based on the overall results of the three-dimensional multiple eNodeB simulations conducted, it is safe to assume that a UE within a LTE network with three eNodeBs can be accurately geolocated within 50 centimeters of its true location based upon known TA data and eNodeB heights and locations. Networks with four eNodeBs provide geolocation accuracy of a LTE UE within 100 meters. LTE UEs can be accurately geolocated in networks of five eNodeBs within 35 meters. Networks with six or more eNodeBs provide LTE UE geolocation accuracy well within 10 meters. Sample plots generated from the three-dimensional multiple eNodeB simulations are presented in Appendix C, including examples of the ideal network results with negligible approximation error, on the level of 10^{-5} meters.

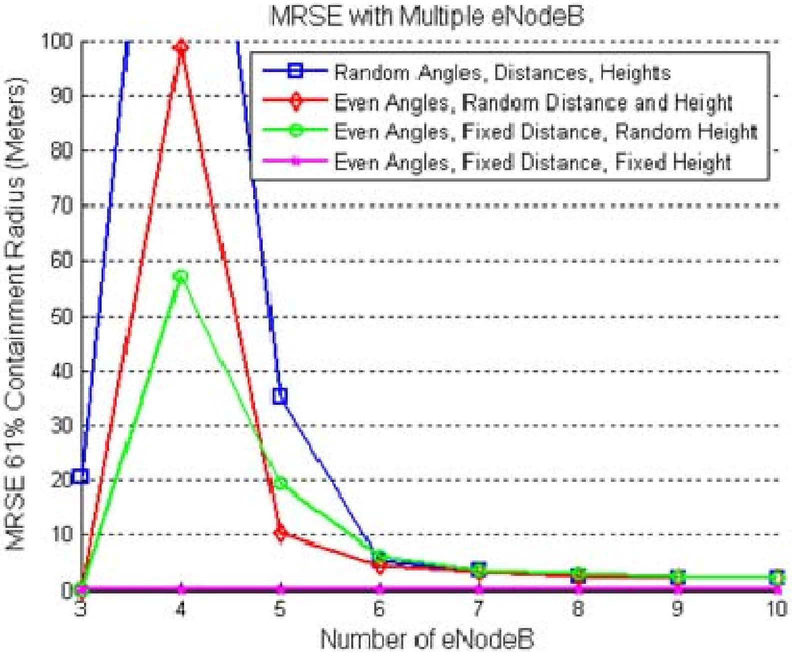


Figure 40. Three-dimensional MRSE error analysis with multiple eNodeBs, standard deviation applied and disregarding approximation Z-coordinate in the four eNodeB network (enhanced view).

Geolocating an LTE UE in a three-dimensional aspect based on the propagation of LTE signals containing TA information transmitting from a variable height of an eNodeB to a UE was explored in this chapter. With calculated values of TA established in Chapter II, position approximation algorithms developed in Chapter III were refined to create a three-dimensional computer simulation model that accurately located a UE using spherical systems of equations. Monte Carlo simulation schemes found that using average distance from an approximated position to actual UE location or MRSE as a measure, an LTE UE can consistently be located within the distance per unit of TA, less than 50 centimeters from the actual position in a three eNodeB network and less than 35 meters from the actual position in networks of five or more eNodeBs. The three-dimensional simulation results showed excellent geolocation capability and proved to be a far more accurate solution in comparison to a two-dimensional form of geolocation explored in Chapter III.

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

Extraction of TA data from an initial RAR message is a viable means of correlating range radius between an LTE eNodeB and UE. With the use of multiple range radii from several eNodeB, crossing radii can very effectively establish a UE location.

Computer simulations modeling an LTE network showed promising geolocation results based upon resolved per unit TA derived from an 11-bit MAC field and little to no variance. The use of trilateration of triangles on a two-dimensional plane showed geolocation accuracy should consistently be practical within a 60-meter CEP. Results improve exponentially with trilateration of spheres on a three-dimensional plane, which indicate a consistent geolocation accuracy practical within 50 centimeters MRSE for a three eNodeB network and within 35 meters MRSE for network sizes of five or more eNodeBs. While it is possible to have very approximate location estimates in a four eNodeB network, variance due to ambiguity in the results suggest a higher degree of inaccuracy when compared to other network sizes for geolocation purposes.

The prerequisites of such accurate location approximations reside with specific knowledge of TA values and offsets and eNodeB tower characteristics of location and height above the ground. This information is imperative and important to consider when employing a system capable of geolocation. Any error in tower characteristics has potential to offset geolocation results since intersections of radii based on that tower, whether circular or spherical, will increase the distance between the UE estimate and true location.

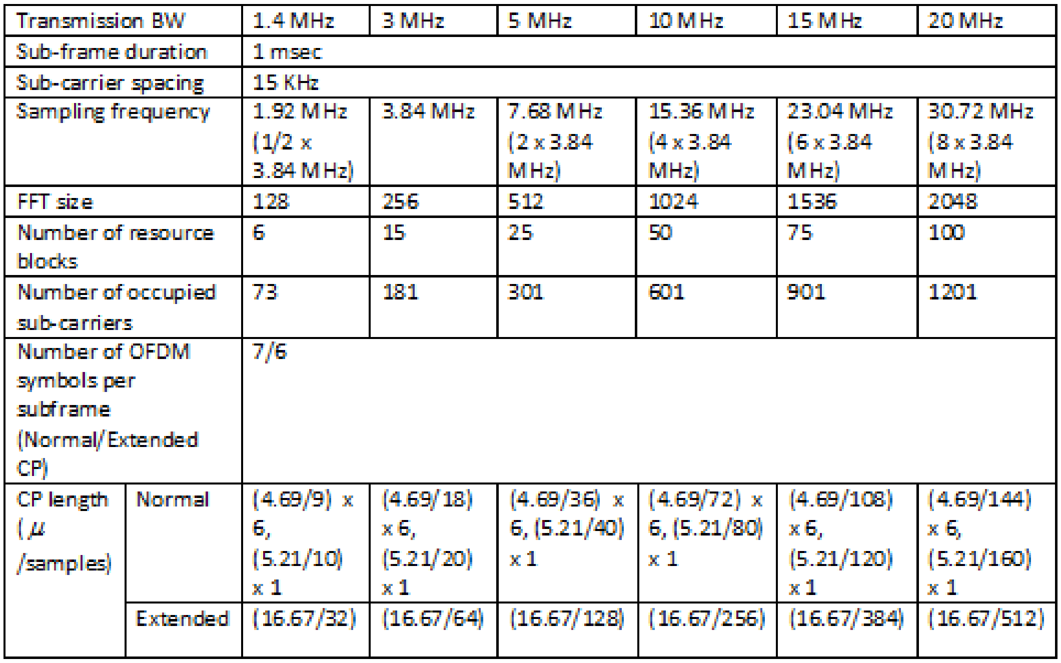
With the principles explored through research and simulation, it is possible to geolocate an LTE UE within approximately 60 meters when using a two-dimensional mapping coordinate mapping scheme, offering potentially ten times better accuracy than GSM methods previously explored in literature. Furthermore, it is possible to geolocate an LTE UE within approximately 50 centimeters by using a three-dimensional mapping approach, taking into account that TA is a calculated distance from one height to another, and is comparable to geolocation accuracy in GPS technologies. Both geolocation methods provide an excellent resource to be refined and expanded upon for employment by emergency response teams and tactical personnel. The current deployment of LTE as a 4G network for the two largest mobile carriers demands effective means of geolocation, such as those presented in this research.

B. RECOMMENDATIONS

Future research should concentrate primarily on successful extraction of LTE signals from the air and analysis of TA data and real-world LTE signal internals behavior. Field testing and collection of real-world data can be used to validate and refine the findings in this research. Development of a software graphical user interface that employs the geolocation methods established for LTE would be a valuable commodity for emergency response teams and tactical users operating in areas with currently deployed or soon-to-be deployed networks. Augmentation of the three-dimensional multiple eNodeB simulation could provide less approximation error by comparing and contrasting location estimate results from individual groups of three eNodeBs within the larger network. A thorough investigation into the LTE specifications for other aspects and mechanisms of ranging should be conducted, along with a focus on the TDD functionality and eNodeB switching capability between duplexing schemes.

APPENDIX A. LTE PARAMETERS FOR DOWNLINK TRANSMISSION SCHEME

Table 1. Parameters for Downlink Transmission Scheme (After [12]).



Transmission BW		1.4 MHz	3 MHz	5 MHz	10 MHz	15 MHz	20 MHz
Sub-frame duration		1 msec					
Sub-carrier spacing		15 KHz					
Sampling frequency		1.92 MHz (1/2 x 3.84 MHz)	3.84 MHz	7.68 MHz (2 x 3.84 MHz)	15.36 MHz (4 x 3.84 MHz)	23.04 MHz (6 x 3.84 MHz)	30.72 MHz (8 x 3.84 MHz)
FFT size		128	256	512	1024	1536	2048
Number of resource blocks		6	15	25	50	75	100
Number of occupied sub-carriers		73	181	301	601	901	1201
Number of OFDM symbols per subframe (Normal/Extended CP)		7/6					
CP length (μ /samples)	Normal	(4.69/9) x 6, (5.21/10) x 1	(4.69/18) x 6, (5.21/20) x 1	(4.69/36) x 6, (5.21/40) x 1	(4.69/72) x 6, (5.21/80) x 1	(4.69/108) x 6, (5.21/120) x 1	(4.69/144) x 6, (5.21/160) x 1
	Extended	(16.67/32)	(16.67/64)	(16.67/128)	(16.67/256)	(16.67/384)	(16.67/512)

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. TWO-DIMENSIONAL SIMULATIONS

A. TWO BASE STATIONS THROUGH VARYING ANGLES

1. Two Base Stations Through Varying Angles MATLAB Code

```
% LTE-Two Base Station Simulation with Varying Angle
% LT L. A. Jarvis, NPS GSEAS
% Rev. 7, 20 Nov 2010

clear all;

% LTE Calculated Timing Advance Distance Parameter
TA = 78.125;    %Meters per unit of Timing Advance

% Uncomment for applying Timing Advance standard deviation
%stdTA = 0.5;

% Uncomment for applying Quantization error based on TA
%qError = 78.125;

% Uncomment for Bias based on quantization error
%bias = qError/2;

% Simulated eNodeB Distance Parameters
siteDist = 1000;   %Mean Site Distance
siteStDev = 300;   %Site Standard Deviation

% Simulation Parameters
iterate = 100000;

% Circle Plots
numPoints = 30; %Number of points in circle plots
rad = (0:numPoints)*2*pi/numPoints;
phi = 18;

% Initialize arrays for average angle and distance error
angleAverage = 0:phi;       %Angle Average
distanceError = 0:iterate;  %Distance Error
degrees = 0:phi;

% Iterate angles in 10 degree increments
for i = 0:phi
    theta = i*pi/phi; %Angle from Subscriber to eNodeB 2
    degrees(i+1) = theta*180/pi;

    % Repeated iterations for averaging
    for j = 1:iterate
        % Distance to simulated eNodeBs with (X,Y) coordinates
% eNodeB on axis
Dist_eNodeB_1 = siteStDev*randn(1) + siteDist;
x_1 = Dist_eNodeB_1;
y_1 = 0;

% Second eNodeB
Dist_eNodeB_2 = siteStDev*randn(1) + siteDist;
x_2 = cos(theta)*Dist_eNodeB_2;
y_2 = sin(theta)*Dist_eNodeB_2;

% Timing Advance for eNodeBs
TA_eNodeB_1 = round(Dist_eNodeB_1/TA);
TA_eNodeB_2 = round(Dist_eNodeB_2/TA);
% Uncomment for applying Timing Advance standard deviation
%TA_eNodeB_1 = round(Dist_eNodeB_1/TA)+round(stdTA*randn(1));
%TA_eNodeB_2 = round(Dist_eNodeB_2/TA)+round(stdTA*randn(1));

% Site Radii based on TA
radii_eNodeB_1 = TA_eNodeB_1 * TA;
radii_eNodeB_2 = TA_eNodeB_2 * TA;
% Uncomment for applying quantization error
%radii_eNodeB_1 = TA_eNodeB_1 * TA + qError*rand(1)-bias;
%radii_eNodeB_2 = TA_eNodeB_2 * TA + qError*rand(1)-bias;

% Intersite distance via Pythagorean Theorem and angle from
% ATAN2.  Intersite distance will be random through each
% iteration
intersiteDist = sqrt((x_2-x_1)^2 + (y_2-y_1)^2);
angle = atan2((y_2-y_1),(x_2-x_1));

% Find Midpoint Between Circles or Circle Intersections
% Separated Radii
if(intersiteDist > radii_eNodeB_1 + radii_eNodeB_2)
    delta_R = intersiteDist - radii_eNodeB_1 - radii_eNodeB_2;
    midpoint_X = x_1+cos(angle)*(radii_eNodeB_1+0.5*delta_R);
    midpoint_Y = y_1+sin(angle)*(radii_eNodeB_1+0.5*delta_R);
% Contained Circles
elseif (intersiteDist < abs(radii_eNodeB_1 - radii_eNodeB_2))
    if(radii_eNodeB_1 > radii_eNodeB_2)
        delta_R = radii_eNodeB_1 - radii_eNodeB_2 -...
            intersiteDist;
        midpoint_X = x_1+cos(angle)*(radii_eNodeB_1-...
            0.5*delta_R);
        midpoint_Y = y_1+sin(angle)*(radii_eNodeB_1-...
            0.5*delta_R);
    else
        delta_R = radii_eNodeB_2 - radii_eNodeB_1 -...
            intersiteDist;
        midpoint_X = x_1+cos(angle)*(-radii_eNodeB_1-...
            0.5*delta_R);
        midpoint_Y = y_1+sin(angle)*(-radii_eNodeB_1-...
            0.5*delta_R);
    end
% Intersections
```

```
        else
            % Distance eNodeB_1 to Midpoint
            midpoint_eNodeB_1 =(radii_eNodeB_1*radii_eNodeB_1-...
                radii_eNodeB_2*radii_eNodeB_2+intersiteDist*...
                intersiteDist)/(2*intersiteDist);
            midpoint_X=x_1+midpoint_eNodeB_1*(x_2-x_1)/intersiteDist;
            midpoint_Y=y_1+midpoint_eNodeB_1*(y_2-y_1)/intersiteDist;
        end
        % Distance from Midpoint to Subscriber at Origin
        distanceError(j) = sqrt(midpoint_X^2+midpoint_Y^2);
    end
    angleAverage(i+1)=mean(distanceError);

    % Plot Once Every Other Angle
    if mod(i,2)==0
        figure(i/2+1);
        clf;
        hold on;
        axis equal;
        title(['eNodeB 1 on X Axis, eNodeB 2 at ',...
            int2str(i*10), ' Degrees'],'FontSize',12);
        xlabel('Meters','FontSize',12);
        ylabel('Meters','FontSize',12);
        plot(radii_eNodeB_1*cos(rad)+ x_1, radii_eNodeB_1...
            *sin(rad)+ y_1,'-b');
        plot(radii_eNodeB_2*cos(rad)+ x_2, radii_eNodeB_2...
            *sin(rad)+ y_2,'-m');
        plot(x_1,y_1,'^b','MarkerSize',12);
        plot(x_2,y_2,'^m','MarkerSize',12);
        plot(midpoint_X,midpoint_Y,'rp','MarkerSize',12);
        plot(0,0,'sk','MarkerSize',12);
        hold off;
    end
end

% Plot Overall Results Angle Between eNodeB vs. Distance to Midpoint
figure(i/2+2);
clf;
axes1 = axes('Parent',figure(i/2+2),'YGrid','on','XDir','reverse');
hold(axes1,'all');
title('2 eNodeB BS Estimate Error Varying Angle','FontSize',12);
xlabel('Angle Between eNodeBs (Degrees)','FontSize',12);
ylabel(['Average Distance from Center Point Estimate to ',...
    'Subscriber (meters)'],'FontSize',12);
plot(degrees,angleAverage,'MarkerFaceColor',[1 0 0],...
    'MarkerSize',10,'Marker','o','LineWidth',2,'Color',[0 0 1]);
```

2. Two Base Stations Through Varying Angles Example Plots

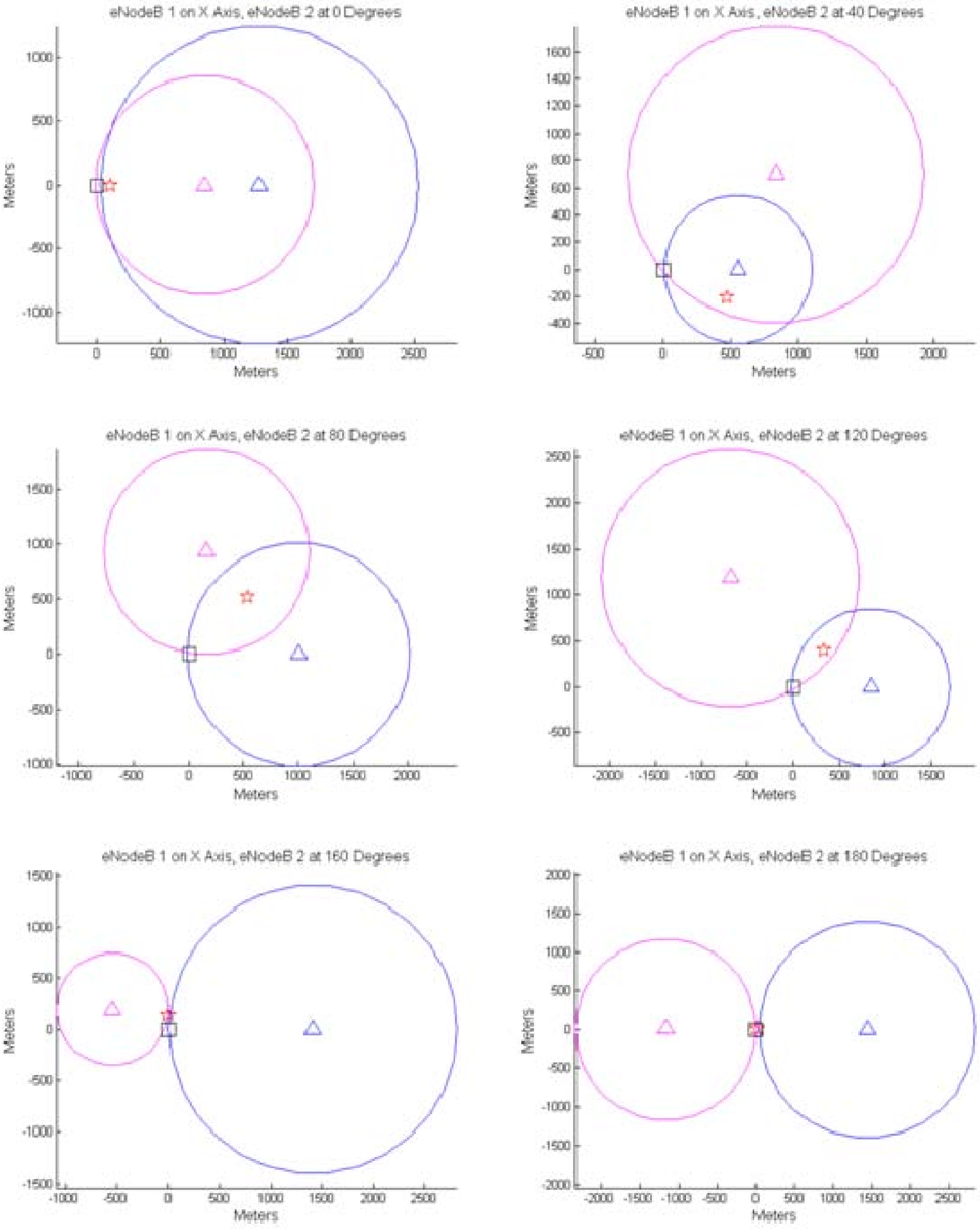


Figure 41. Sample plots from 2 eNodeB simulation with varying angles and distances.

(Triangles denote the eNodeBs, a square indicates the actual UE location, and the star shows approximation center point.)

B. MULTIPLE BASE STATIONS

1. Multiple Base Stations MATLAB Code

```
% Multiple Base Station Simulation
% Combined
% Random Angle, Random Distance (Figures 1-9)
% Even Angle, Random Distance (Figures 10-18)
% Even Angle, Fixed Distance (Figures 19-27)
% Error Estimation Analysis (Figure 28)
% CEP Estimation Analysis (Figure 29)
% LT L. A. Jarvis, NPS GSEAS
% Rev. 12, 29 Nov. 2010

clear all;

% LTE Calculated Timing Advance Distance Parameter
TA = 78.125;    %Meters per unit of Timing Advance

% Uncomment for applying Timing Advance standard deviation
%stdTA = 0.5;

% Quantization error based on TA
%qError = 78.125;

% Bias based on quantization error
%bias = qError/2;

% Simulated eNodeB Distance Parameters
siteDist = 1200;    %Mean Site Distance
siteStDev = 400;    %Site Standard Deviation

% Simulation Parameters
max_eNodeB = 10;    %Maximum number of eNodeB to simulate
iterate = 100000;   %Number of iterations at each eNodeB

% Circle Plots
numPoints = 180;    %Number of points in circle plots
rad = (0:numPoints)*2*pi/numPoints;

% Initialize arrays for data analysis
averageError1 = 0:max_eNodeB-2; %Average Error at N eNodeB
averageError2 = 0:max_eNodeB-2;
averageError3 = 0:max_eNodeB-2;
stDevError1 = 0:max_eNodeB-2;   %Standard Deviation at N eNodeB
stDevError2 = 0:max_eNodeB-2;
stDevError3 = 0:max_eNodeB-2;
CEP1 = 0:max_eNodeB-2;          %Circular Error Probable at N eNodeB
CEP2 = 0:max_eNodeB-2;
CEP3 = 0:max_eNodeB-2;
numBS = 0:max_eNodeB-2;         %Array of number of eNodeB
```

```
% Initialize counter for figures
count = 1;

for D = 1:3
% Setup simulation
for N = 2:max_eNodeB
    % Initialize sized arrays
    distance = zeros(1,N);  %Array of distances to eNodeBs
    siteX = zeros(1,N);     %Array of eNodeB X coordinates
    siteY = zeros(1,N);     %Array of eNodeB Y coordinates
    siteTA = zeros(1,N);    %Array of Timing Advance
    siteRad = zeros(1,N);   %Array of eNodeB radii
    inter1X = zeros(1,N);   %Array of intercept 1 X coordinates
    inter1Y = zeros(1,N);   %Array of intercept 1 Y coordinates
    inter2X = zeros(1,N);   %Array of intercept 2 X coordinates
    inter2Y = zeros(1,N);   %Array of intercept 2 Y coordinates
    % Array of select intercept from pair's X coord
    interX = zeros(1,N);
    % Array of select intercept from pair's Y coord
    interY = zeros(1,N);
    % Array of distances for center point estimation
    decp = zeros(1,iterate);

    % Begin iterations
    for h = 1:iterate
        % Create eNodeBs
        for i = 1:N %Number of eNodeB
            % Distances to simulated eNodeBs with (X,Y) coordinates
            if (D==1)
                distance(i) = siteStDev*randn(1) + siteDist;
                theta = 2*pi*rand(1);   %Random angle from UE
            elseif (D==2)
                distance(i) = siteStDev*randn(1) + siteDist;
                theta = 2*pi*i/N;   %Random angle from UE
            elseif (D==3)
                distance(i) = 1000; %Fixed distances of 1000m
                theta = 2*pi*i/N; %Evenly spaced angles from UE
            end
            siteX(i) = cos(theta)*distance(i);  %X coordinate
            siteY(i) = sin(theta)*distance(i);  %Y coordinate
            %Timing Advance
            siteTA(i) = round(distance(i)/TA);
            % Uncomment for applying Timing Advance standard deviation
            %siteTA(i) = round(distance(i)/TA)+round(stdTA*randn(1));
            %Radius from Timing Advance
            siteRad(i) = siteTA(i) * TA;
            % Uncomment for aplling quantization error
            %siteRad(i) = siteTA(i) * TA + qError*rand(1)-bias;
        end

        % Find Two Intersections for Each Pair of Radii
        for a = 1:N
            b = mod(a,N)+1;

            % Intersite distance via Pythagorean Theorem,
            % angle from ATAN2
            isDist = sqrt((siteX(b)-siteX(a))^2 + (siteY(b)-...
                siteY(a))^2);
            angle = atan2((siteY(b)-siteY(a)),(siteX(b)-siteX(a)));

            % Find Midpoint Between Circles or Circle Intersections
            % Separated Radii
            if(isDist > siteRad(a)+siteRad(b))
                % Closest Point Set as Both Intersections
                deltaR=isDist-siteRad(a)-siteRad(b);
                inter1X(a)=siteX(a)+cos(angle)*(siteRad(a)+0.5*deltaR);
                inter1Y(a)=siteY(a)+sin(angle)*(siteRad(a)+0.5*deltaR);
                inter2X(a)=inter1X(a);
                inter2Y(a)=inter1Y(a);
            % Contained Circles
            elseif (isDist < abs(siteRad(a)-siteRad(b)))
                % Narrowest Set as Both Intersections
                if(siteRad(a) > siteRad(b))
                   deltaR=siteRad(a)-siteRad(b)-isDist;
                   inter1X(a)=siteX(a)+cos(angle)*(siteRad(a)-...
                       0.5*deltaR);
                   inter1Y(a)=siteY(a)+sin(angle)*(siteRad(a)-...
                       0.5*deltaR);
                   inter2X(a)=inter1X(a);
                   inter2Y(a)=inter1Y(a);
                else
                   deltaR=siteRad(b)-siteRad(a)-isDist;
                   inter1X(a)=siteX(a)+cos(angle)*(-siteRad(a)-...
                       0.5*deltaR);
                   inter1Y(a)=siteY(a)+sin(angle)*(-siteRad(a)-...
                       0.5*deltaR);
                   inter2X(a)=inter1X(a);
                   inter2Y(a)=inter1Y(a);
                end
            % Intersections
            else
                % Distance from eNodeB 1 to midpoint
                site1mp=(siteRad(a)^2-siteRad(b)^2+isDist^2)/...
                    (2*isDist);
                % Distance from midpoint to intercept
                mp2inter=sqrt(siteRad(a)^2-site1mp^2);
                % Midpoint X coordinate
                mpX=siteX(a)+site1mp*(siteX(b)-siteX(a))/isDist;
                % Midpoint Y Coordinate
                mpY=siteY(a)+site1mp*(siteY(b)-siteY(a))/isDist;
                % First Intersection Coordinates
                inter1X(a)=mpX+mp2inter*(siteY(b)-siteY(a))/isDist;
                inter1Y(a)=mpY-mp2inter*(siteX(b)-siteX(a))/isDist;
                % Second Intersection Coordinates
                inter2X(a)=mpX-mp2inter*(siteY(b)-siteY(a))/isDist;
                inter2Y(a)=mpY+mp2inter*(siteX(b)-siteX(a))/isDist;
            end
        end

        % Select Closest Proximity Intersection from Each Pair
```

```
    for j = 1:N
        k = mod(j+1,N)+1;
        d1=sqrt((inter1X(k)-inter1X(j))^2+(inter1Y(k)-...
            inter1Y(j))^2);
        d12=sqrt((inter2X(k)-inter1X(j))^2+(inter2Y(k)-...
            inter1Y(j))^2);
        if (d12 < d1)
            d1 = d12;
        end
        d2=sqrt((inter1X(k)-inter2X(j))^2+(inter1Y(k)-...
            inter2Y(j))^2);
        d22=sqrt((inter2X(k)-inter2X(j))^2+(inter2Y(k)-...
            inter2Y(j))^2);
        if (d22 < d2)
            d2 = d22;
        end
        % Store Selected Intersection in Array
        if (d1 < d2)
            interX(j) = inter1X(j); interY(j) = inter1Y(j);
        else
            interX(j) = inter2X(j); interY(j) = inter2Y(j);
        end
    end

    % Approximate Center of Polygon from Selected Intersections
    % by Coordinate Mean
    approxX = mean(interX);
    approxY = mean(interY);

    % Distance from Subscriber to Estimate
    decp(h) = sqrt(approxX^2+approxY^2);
end

if (D==1)
    % Mean Estimate Error at N Stations
    averageError1(N-1) = mean(decp);
    % Standard Deviation of Error
    stDevError1(N-1) = std(decp);
    % Circular Error Prob at N Stations
    CEP1(N-1) = median(decp);
elseif (D==2)
    % Mean Estimate Error at N Stations
    averageError2(N-1) = mean(decp);
    % Standard Deviation of Error
    stDevError2(N-1) = std(decp);
    % Circular Error Prob at N Stations
    CEP2(N-1) = median(decp);
elseif (D==3)
    % Mean Estimate Error at N Stations
    averageError3(N-1) = mean(decp);
    % Standard Deviation of Error
    stDevError3(N-1) = std(decp);
    % Circular Error Prob at N Stations
    CEP3(N-1) = median(decp);
end
```

```
    numBS(N-1) = N;

    % Plot
    figure(count)
    clf;
    hold on;
    axis equal;
    title([int2str(N), ' eNodeB, ' int2str(decp(h)), ...
        ' Meters from Approximation to UE'],'FontSize',12);
    xlabel('Meters','FontSize',12);
    ylabel('Meters','FontSize',12);
    for p = 1:N
        % Plot eNodeB
        plot(siteX(p),siteY(p),'^b','MarkerSize',10);
        % Plot TA Radius
        plot(siteRad(p)*cos(rad)+ siteX(p), siteRad(p)*sin(rad)+...
            siteY(p));
        % Plot Both Intercepts
        plot(inter1X(p),inter1Y(p),'ob','MarkerSize',6);
        plot(inter2X(p),inter2Y(p),'ob','MarkerSize',6);
        % Mark Selected Intercepts
        plot(interX(p),interY(p),'rd','MarkerSize',12);
    end
    % Plot UE
    plot(0,0,'sk','MarkerSize',12);
    % Plot Approximation
    plot(approxX,approxY,'rp','MarkerSize',12);
    hold off;

    count = count+1;
end
end

% Plot Overall Results Number of eNodeB vs. Distance to Midpoint
figure(count);
clf;
hold on;
title('Estimate Error with Multiple eNodeB','FontSize',12);
xlabel('Number of eNodeB','FontSize',12);
ylabel({'Average Distance from';
    'Center Point Estimate to UE (meters)'},'FontSize',12);
plot(numBS(2:N-1),averageError1(2:N-1),'-bs','LineWidth',2);
plot(numBS(2:N-1),averageError2(2:N-1),'-rd','LineWidth',2);
plot(numBS(2:N-1),averageError3(2:N-1),'-go','LineWidth',2);
legend('Random Angle & Distances','Even Angles, Random Distances',...
    'Even angles, all 1 km','FontSize',12);
hold off;

% Plot CEP Overall Results Number of eNodeB vs. Distance to Midpoint
figure(count+1);
clf;
hold on;
title('CEP Estimate Error with Multiple eNodeB','FontSize',12);
xlabel('Number of eNodeB','FontSize',12);
ylabel('CEP Radius (Meters)','FontSize',12);
```

```
plot(numBS(2:N-1),CEP1(2:N-1),'-bs','LineWidth',2);
plot(numBS(2:N-1),CEP2(2:N-1),'-rd','LineWidth',2);
plot(numBS(2:N-1),CEP3(2:N-1),'-go','LineWidth',2);
legend('Random Angle & Distances','Even Angles, Random Distances',...
    'Even angles, all 1 km','FontSize',12);
hold off;
```

2. Random Angle and Distance Example Plots

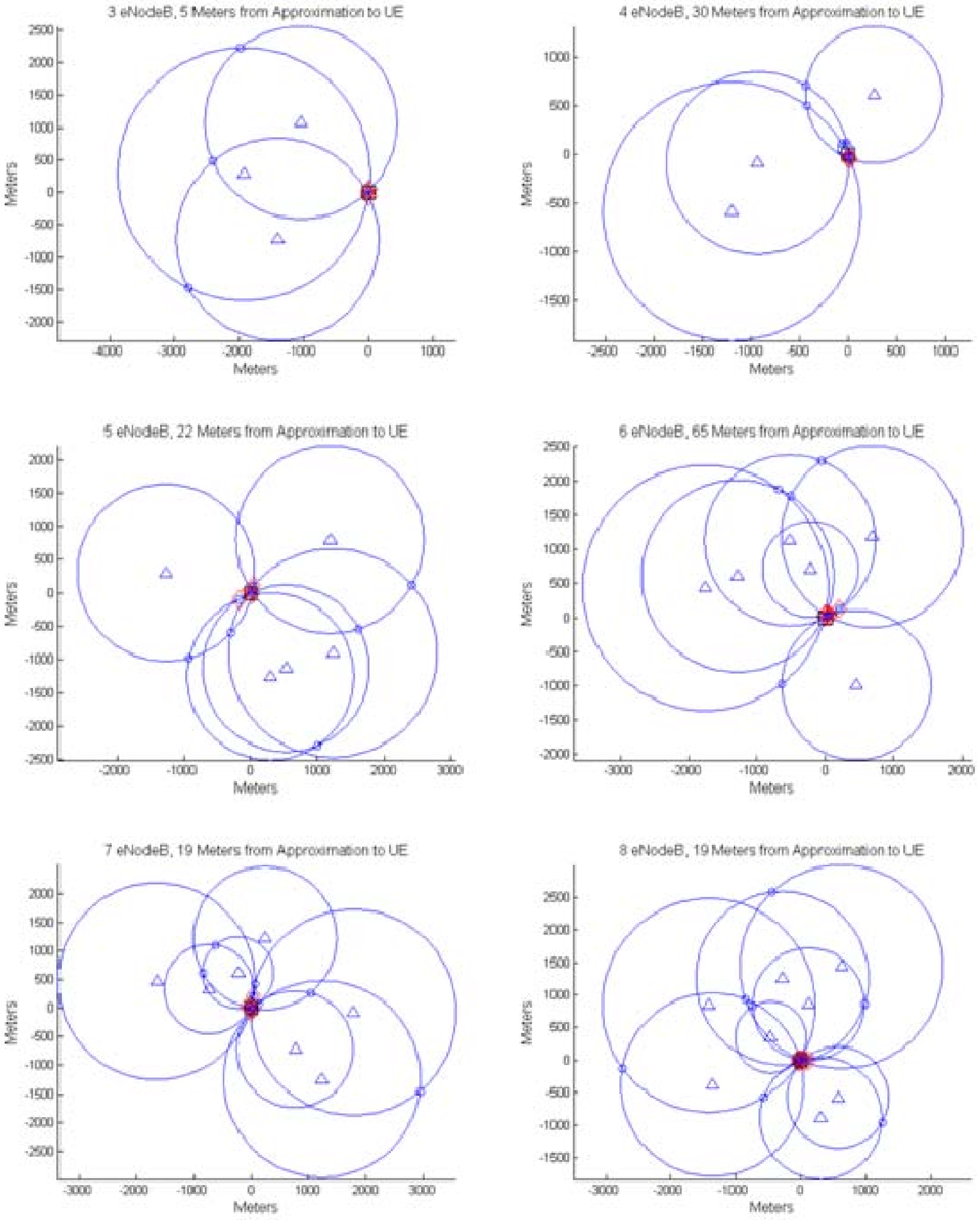


Figure 42. Sample multiple eNodeB plots with random angles and distances.

(Triangles denote the eNodeBs, a square indicates the actual UE location, and the star shows approximation center point.)

3. Evenly Spaced Angles with Random Distance Example Plots

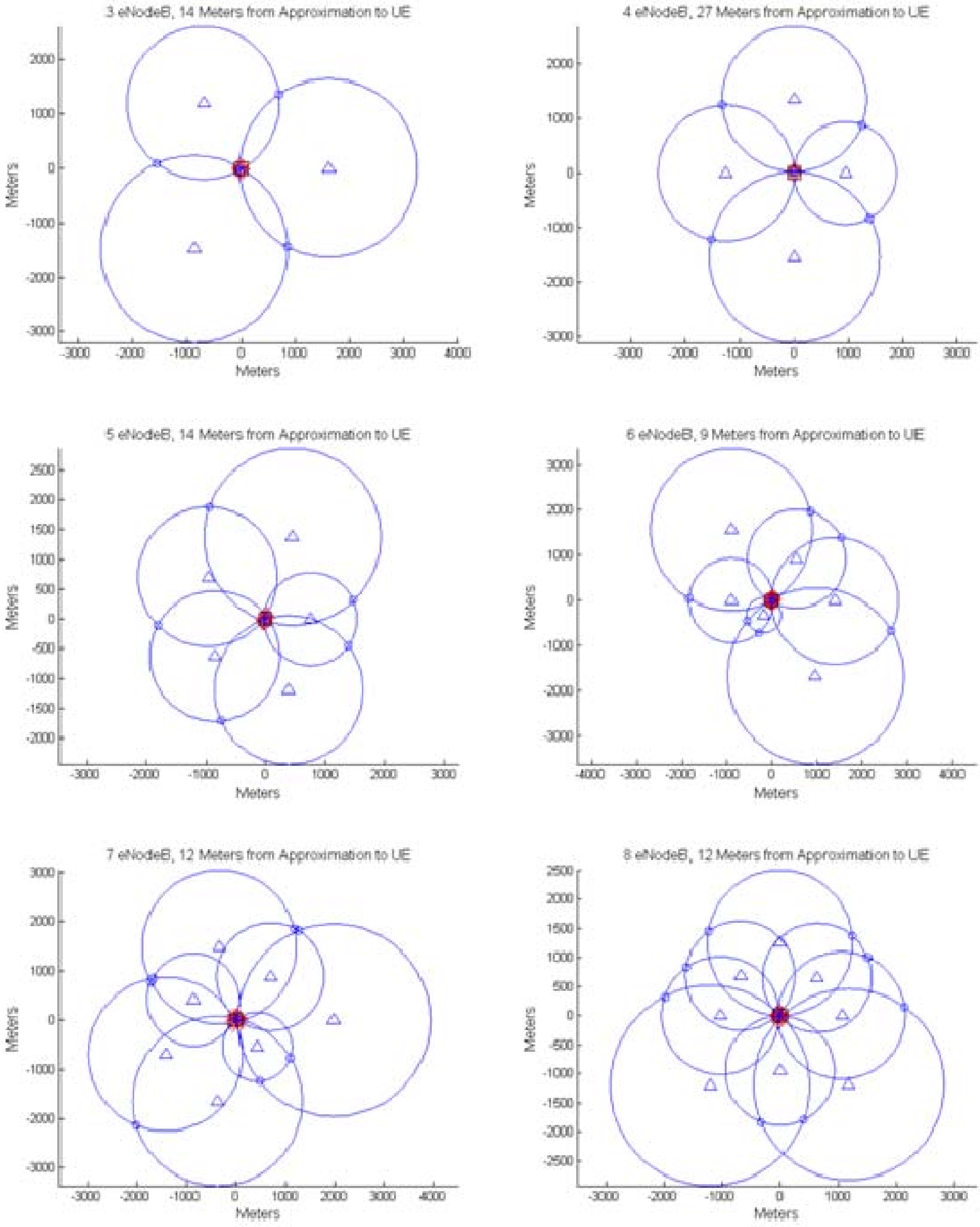


Figure 43. Sample multiple eNodeB plots with even angles and random distances.

(Triangles denote the eNodeBs, a square indicates the actual UE location, and the star shows approximation center point.)

4. Evenly Spaced Angles with Fixed Distance Example Plots

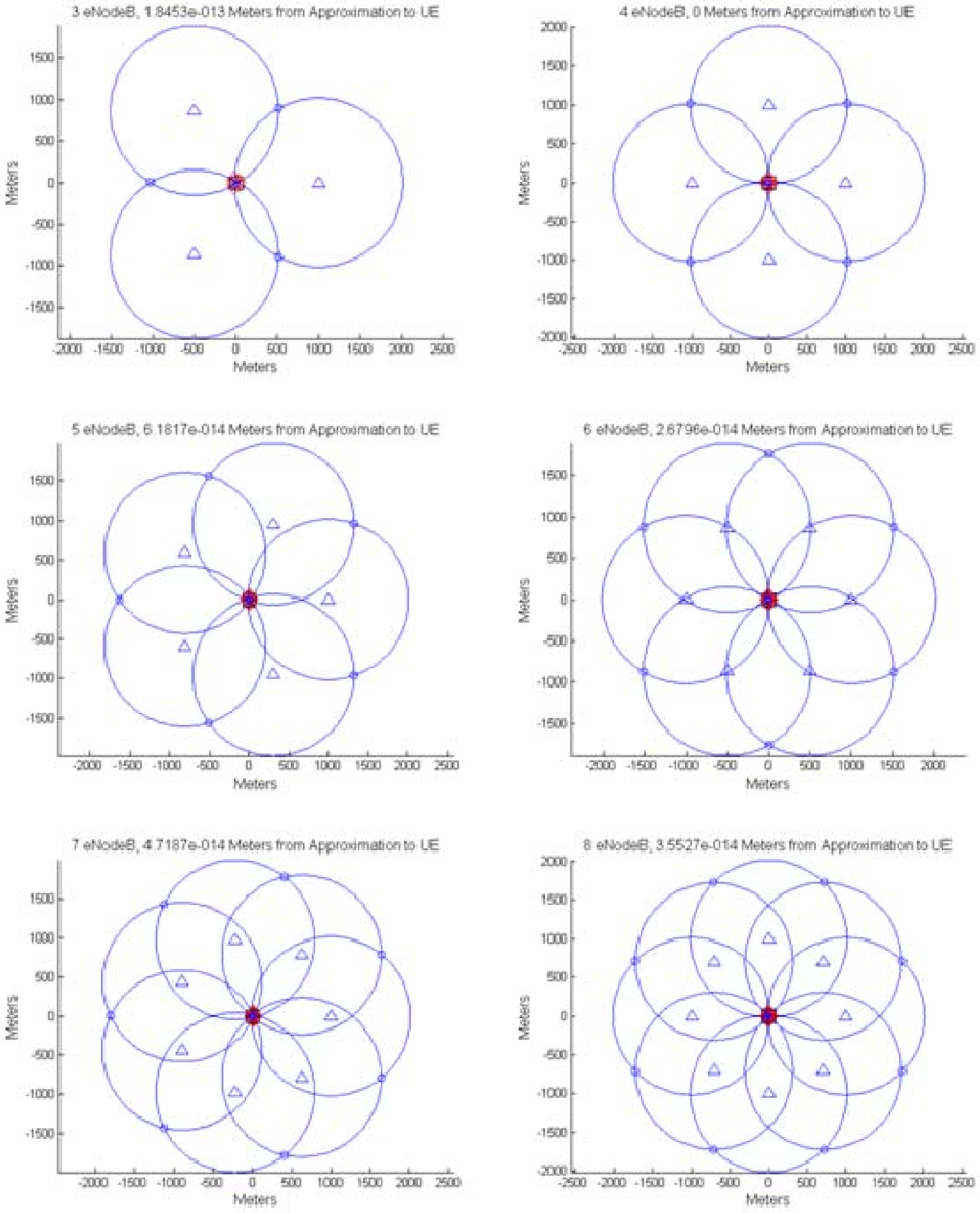


Figure 44. Sample multiple eNodeB plots with even angles and fixed 1 km distance.

(Triangles denote the eNodeBs, a square indicates the actual UE location, and the star shows approximation center point.)

5. Evenly Spaced Angles with Fixed Distance Example Plots, Standard Deviation Applied

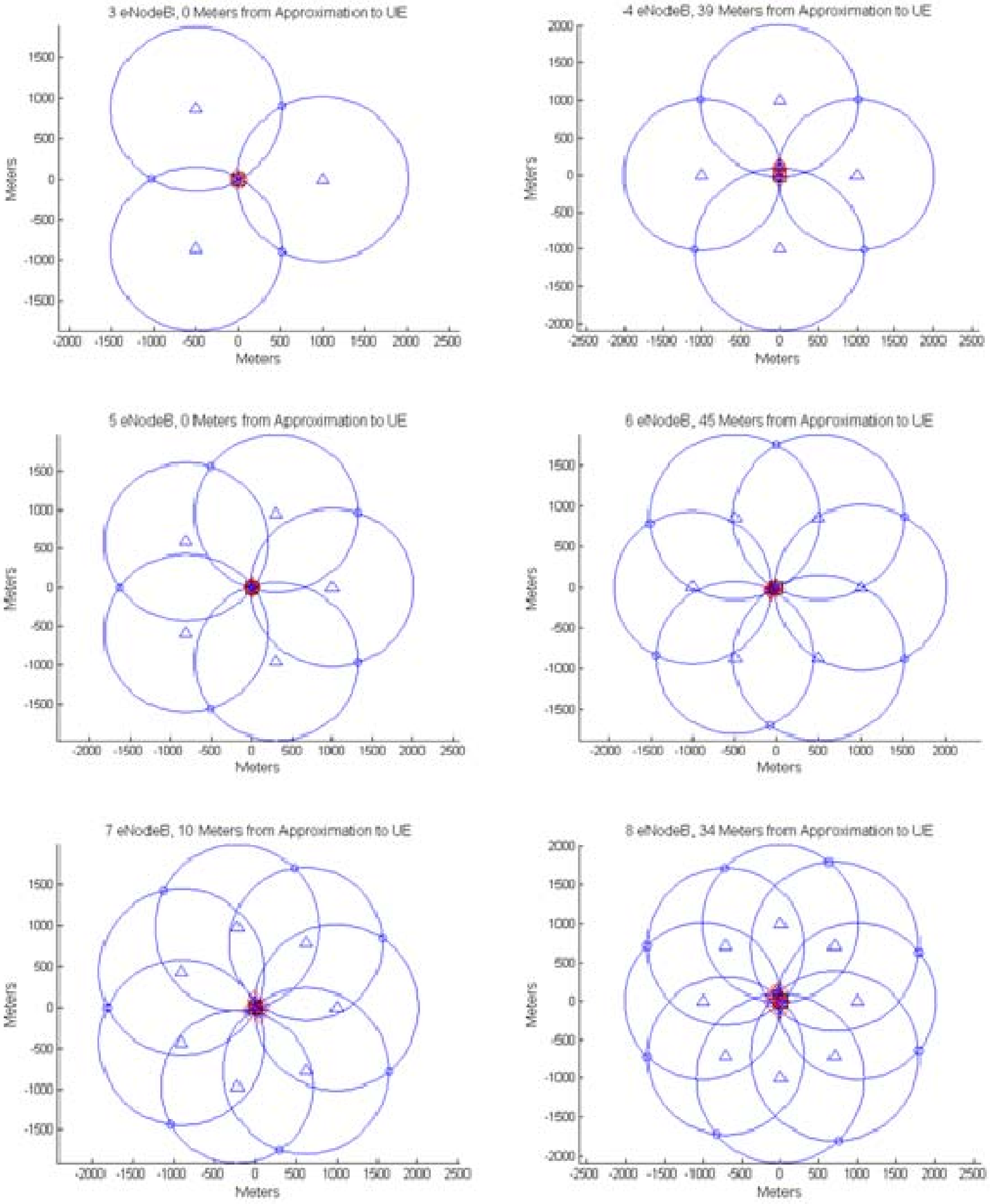


Figure 45. Sample multiple eNodeB plots with even angles and fixed 1 km distance, standard deviation applied.

APPENDIX C. THREE-DIMENSIONAL SIMULATIONS

A. THREE BASE STATIONS

1. Three Base Stations Through Varying Angle MATLAB Code

```
% 3D Base Station Simulation
% Three eNodeBs
% Random Angles, Random Distances
% LT L. A. Jarvis
% Rev. 9, 29 November 2010

clear all;

% LTE Calculated Timing Advance Distance Parameter
TA = 78.125;    %Meters per unit of Timing Advance
% Uncomment to apply Timing Advance standard deviation
%stdTA = 0.5;
% Uncomment to apply quantization error standard deviation
%qError = 78.125;
% Uncomment to apply bias from quantization error
%bias = qError/2;

% Simulated eNodeB Distance Parameters
siteDist = 1200;    %Mean Site Distance
siteStDev = 400;    %Site Standard Deviation
siteHigh = 305;     %Mean Site Height
siteHSD = 250;      %Site Height Standard Deviation

% Simulation Parameters
numSimulation = 10;
max_eNodeB = 3;
iterate = 100000;

% Sphere Plots
numPoints = 10;
phi = 0:pi/numPoints:2*pi;
th =(0:pi/numPoints:pi)';


% Initialize arrays for data analysis
% Average error per simulation
averagePositionError = zeros(1,numSimulation);
% Standard dev per simulation
stDevPositionError = zeros(1,numSimulation);
% Mean Radial Spherical Containment
MRSE = zeros(1,numSimulation);
%Simulation Counter
numSim = zeros(1,numSimulation);
```

```
for N=1:numSimulation
    % Initialize sized arrays
    distance = zeros(1,max_eNodeB); %Array of distances to eNodeBs
    siteX = zeros(1,max_eNodeB);    %Array of eNodeB X coordinates
    siteY = zeros(1,max_eNodeB);    %Array of eNodeB Y coordinates
    siteZ = zeros(1,max_eNodeB);    %Array of eNodeB Z coordinates
    siteTA = zeros(1,max_eNodeB);   %Array of Timing Advance
    siteRad = zeros(1,max_eNodeB);  %Array of eNodeB radii
    interX = zeros(1,iterate);
    interY = zeros(1,iterate);
    interZ = zeros(1,iterate);
    % Array of distances for center point estimation on (x,y,z) plane
    decp = zeros(1,iterate);

    for h=1:iterate
        % Create eNodeBs
        for i=1:max_eNodeB
            % Distances to simulated eNodeBs with (X,Y,Z) coordinates
            distance(i) = siteStDev*randn(1) + siteDist;
            theta = 2*pi*rand(1);                 %Random angle from UE
            siteX(i) = cos(theta)*distance(i);            %X coordinate
            siteY(i) = sin(theta)*distance(i);            %Y coordinate
            siteZ(i) = abs(siteHSD*randn(1) + siteHigh);  %Z coordinate
            siteTA(i) = round(distance(i)/TA);          %Timing Advance
            % Uncomment to apply Timing Advance standard deviation
            %siteTA(i) = round(distance(i)/TA)+round(stdTA*randn(1));
            siteRad(i) = siteTA(i) * TA;                %Radius from TA
            % Uncomment to apply quantization error
            %siteRad(i)=siteTA(i) * TA + qError * rand(1) - bias;
        end

        % Find x-variable coefficients
        a11 = (siteX(2)-siteX(1))^2;
        a21 = (siteX(3)-siteX(1))^2;

        % Find y-variable coefficients
        a12 = (siteY(2)-siteY(1))^2;
        a22 = (siteY(3)-siteY(1))^2;

        % Find z-variable coefficients
        a13 = (siteZ(2)-siteZ(1))^2;
        a23 = (siteZ(3)-siteZ(1))^2;

        % Find values for B-matrix
        b1 = siteRad(1)^2-siteRad(2)^2-siteX(1)^2+siteX(2)^2-...
            siteY(1)^2+siteY(2)^2-siteZ(1)^2+siteZ(2)^2;
        b2 = siteRad(1)^2-siteRad(3)^2-siteX(1)^2+siteX(3)^2-...
            siteY(1)^2+siteY(3)^2-siteZ(1)^2+siteZ(3)^2;

        % Coefficient Matrix A
        A = [a11 a12 a13; a21 a22 a23];

        % Matrix B
        B = [b1; b2];

        % Find solutions to x, y, and z
        X=A\B;

        interX(h)=X(1);
        interY(h)=X(2);
        interZ(h)=X(3);

        approxX=mean(interX);
        approxY=mean(interY);
        approxZ=mean(interZ);

        decp(h)=sqrt(approxX^2+approxY^2+approxZ^2);
    end

    % Mean estimate position error per simulation
    averagePositionError(N)=mean(decp);
    % Standard Deviation of Error
    stDevPositionError(N)=std(decp);
    % Mean Radial Spherical Error per simulation
    stdX = std(interX);
    stdY = std(interY);
    stdZ = std(interZ);
    MRSE(N)= sqrt(stdX^2+stdY^2+stdZ^2);
    numSim(N) = N;

    % Plot
    figure(N);
    clf;
    hold on;
    col = [1 0 0, 0 0 1, 0 1 0, 1 1 0]; %Color map
    v = 1;                              %Color counter
    title(['3 eNodeB, ',num2str(decp(h)),' Meters from '...
        'Approximation to UE'],'FontSize',12);
    xlabel('X-Axis (meters)','FontSize',12);
    ylabel('Y-Axis (meters)','FontSize',12);
    zlabel('Height (meters)','FontSize',12);

    for p=1:max_eNodeB
        % Create spheres
        mesh(siteX(p)+siteRad(p)*sin(th)*cos(phi),siteY(p)+...
            siteRad(p)*sin(th)*sin(phi),siteZ(p)+...
            siteRad(p)*cos(th)*ones(size(phi,1),size(phi,2)),...
            'FaceAlpha',0.2,'FaceColor',col(v:v+2),'EdgeColor',...
            col(v:v+2),'EdgeAlpha',0.1);

        % Create antenna center points
        plot3(siteX(p),siteY(p),siteZ(p),'d','MarkerFaceColor',...
            col(v:v+2),'MarkerSize',10);

        % Create towers
        ts=round(siteZ(p));
        tx=zeros(1,ts);
```

```
        ty=zeros(1,ts);
        tz=zeros(1,ts);
        for t=0:ts
            tx(t+1)=siteX(p);
            ty(t+1)=siteY(p);
            tz(t+1)=t;
        end

        % Plot Towers
        plot3(tx,ty,tz,'-k','linewidth',4);

        v = v+3;
    end

    % Plot Approximation Point
    plot3(approxX,approxY,approxZ,'pr','MarkerSize',12);

    % Plot UE
    plot3(0,0,0,'sk','MarkerSize',12);
    hold off;
end

% Plot Overall Results Error of Approximation Distance
% to Actual Position
figure(N+1);
clf;
axes1 = axes('Parent',figure(N+1),'YGrid','on');
hold(axes1,'all');
title('Estimate Distance Error with Three eNodeBs',...
    'FontSize',12);
xlabel('Simulation (each with 100000 iterations)','FontSize',12);
ylabel({'Average Distance from';...
    'Approximated Location to Actual UE (meters)'},'FontSize',12);
bar(numSim,averagePositionError);

% Plot MRSE of position error
figure(N+2);
clf;
axes1 = axes('Parent',figure(N+2),'YGrid','on');
hold(axes1,'all');
title('MRSE with Three eNodeBs','FontSize',12);
xlabel('Simulation (each with 100000 iterations)','FontSize',12);
ylabel('MRSE 61% Containment Radius(meters)','FontSize',12);
bar(numSim,MRSE);
```

2. Three Base Stations Through Varying Angles Example Plots

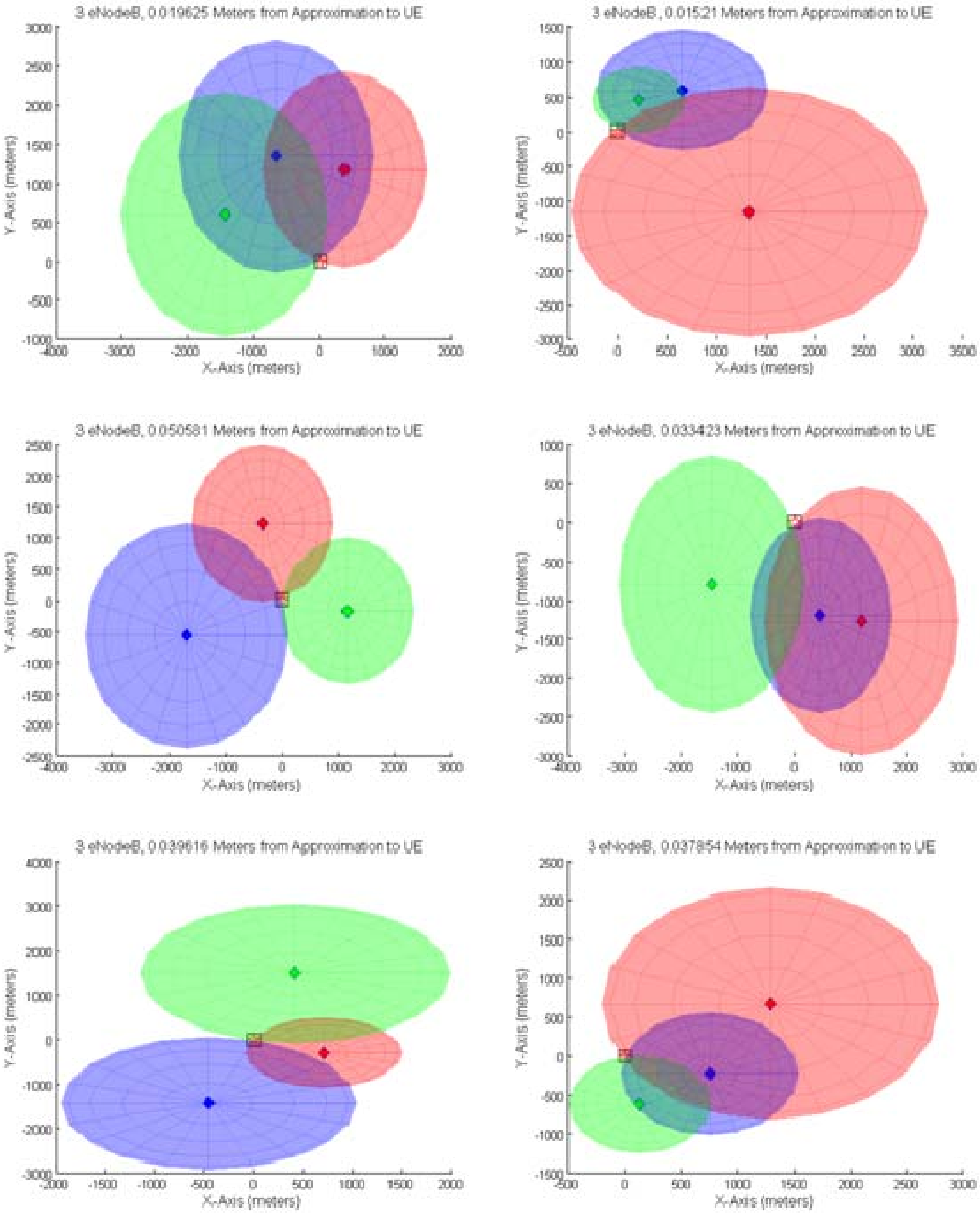


Figure 46. Sample plots from three base station simulation with varying angles, random distances and random eNodeB heights.

(Diamonds denote the eNodeBs, a square indicates the actual UE location, and the star shows approximation center point.)

B. FOUR BASE STATIONS

1. Four Base Stations MATLAB Code

```
% 3D Base Station Simulation
% 4 eNodeBs
% Random Angles, Distances and Heights (Fig 1-10)
% Even Angles, Random Distances and Heights (Fig 11-20)
% Even Angles, Fixed Distance of 1732m, Random Heights (Fig 21-30)
% Even Angles, Fixed Distances and Heights (Fig 31-40)
% Error Estimation Analysis (Fig 41)
% MSRE Error Estimation Analysis (Fig 42)
% LT L. A. Jarvis
% Rev. 7, 29 November 2010

clear all;

% LTE Calculated Timing Advance Distance Parameter
TA = 78.125;    %Meters per unit of Timing Advance
% Uncomment to apply Timing Advance standard deviation
%stdTA = 0.5;
% Uncomment to apply quantization error standard deviation
%qError = 78.125;
% Uncomment to apply bias based on quantization error
%bias = qError/2;

% Simulated eNodeB Distance Parameters
siteDist = 1200;    %Mean Site Distance
siteStDev = 400;    %Site Standard Deviation
siteHigh = 305;     %Mean Site Height
siteHSD = 250;      %Site Height Standard Deviation

% Simulation Parameters
numSimulation = 10;
max_eNodeB = 4;
iterate = 100000;

% Sphere Plots
numPoints = 10;
phi = 0:pi/numPoints:2*pi;
th =(0:pi/numPoints:pi)';


% Initialize arrays for data analysis
% Average error per simulation
averagePositionError1 = zeros(1,numSimulation);
averagePositionError2 = zeros(1,numSimulation);
averagePositionError3 = zeros(1,numSimulation);
averagePositionError4 = zeros(1,numSimulation);
% Standard dev per simulation
stDevPositionError1 = zeros(1,numSimulation);
stDevPositionError2 = zeros(1,numSimulation);
```

```
stDevPositionError3 = zeros(1,numSimulation);
stDevPositionError4 = zeros(1,numSimulation);
% Mean Radial Spherical Error
MRSE1 = zeros(1,numSimulation);
MRSE2 = zeros(1,numSimulation);
MRSE3 = zeros(1,numSimulation);
MRSE4 = zeros(1,numSimulation);
% Simulation Counter
numSim = zeros(1,numSimulation);

% Initialize counter for figures
count = 1;

for D = 1:4
    % Setup Simulation
    for N=1:numSimulation
        % Initialize sized arrays
        distance = zeros(1,max_eNodeB); %Array of distances to eNodeBs
        siteX = zeros(1,max_eNodeB);    %Array of eNodeB X coordinates
        siteY = zeros(1,max_eNodeB);    %Array of eNodeB Y coordinates
        siteZ = zeros(1,max_eNodeB);    %Array of eNodeB Z coordinates
        siteTA = zeros(1,max_eNodeB);   %Array of Timing Advance
        siteRad = zeros(1,max_eNodeB);  %Array of eNodeB radii
        interX = zeros(1,iterate);
        interY = zeros(1,iterate);
        interZ = zeros(1,iterate);
        aX = zeros(1,max_eNodeB-1);       %Array of X coefficients
        aY = zeros(1,max_eNodeB-1);       %Array of Y coefficients
        aZ = zeros(1,max_eNodeB-1);       %Array of Z coefficients
        b = zeros(1,max_eNodeB-1);        %Array of B constants
        % Array of distances for center point estimation on (x,y,z)
  % plane
        decp = zeros(1,iterate);

        % Begin iterations
        for h=1:iterate
            % Create eNodeBs
            for i=1:max_eNodeB
            % Distances to simulated eNodeBs with (X,Y,Z) coordinates
                if (D==1)
                    distance(i) = siteStDev*randn(1) + siteDist;
                    theta = 2*pi*rand(1);       %Random angles from UE
                    % Random eNodeB Height...Z coordinate
                    siteZ(i) = abs(siteHSD*randn(1) + siteHigh);
                elseif (D==2)
                    distance(i) = siteStDev*randn(1) + siteDist;
                    theta = 2*pi*i/max_eNodeB;    %Even angles from UE
                    % Random eNodeB Height...Z coordinate
                    siteZ(i) = abs(siteHSD*randn(1) + siteHigh);
                elseif (D==3)
                    distance(i) = 1732;         %Fixed distances
                    theta = 2*pi*i/max_eNodeB;    %Even angles from UE
                    % Random eNodeB Height...Z coordinate
                    siteZ(i) = abs(siteHSD*randn(1) + siteHigh);
                elseif (D==4)
                    distance(i) = 1732;         %Fixed distances
                    theta = 2*pi*i/max_eNodeB;    %Even angles from UE
                    % Fixed eNodeB Height...Z coordinate
                    siteZ(i) = siteHigh;
                end
                siteX(i) = cos(theta)*distance(i);  %X coordinate
                siteY(i) = sin(theta)*distance(i);  %Y coordinate
                siteTA(i) = round(distance(i)/TA);  %Timing Advance
                % Uncomment to apply Timing Advance standard deviation
                %siteTA(i)=round(distance(i)/TA)+round(stdTA*randn(1));
                siteRad(i) = siteTA(i) * TA;         %Radius from TA
                % Uncomment to apply quantization error to radii
                %siteRad(i) = siteTA(i)*TA+qError*rand(1)-bias;
            end

            % Find point of intersection

            % Due to even angles, fixed distances and fixed heights,
            % this simulation has the potential to find a case where
            % there is no solution or many solutions.  For systems
            % of equations, the unknown variables can be solved using
            % matrices in the form of AX=B, where A contains the
            % coefficients of the unknown variables.  Since the
            % coefficients for x, y and z are the same for all
            % iterations, the only values that need to be determined
            % are the coefficients for matrix B.  Matrix A is
            % singular, meaning the determinant of A is equal to 0.
            % Therefore, AX=B either does not exist, or is not unique.
            % For these cases, a least-squares solution is found
            % using MATLAB function "pinv" so that the simulation can
            % continue to run without interruption, and provide
            % reasonable results.

            % Find x-variable coefficients
            a11 = (siteX(2)-siteX(1))^2;
            a21 = (siteX(3)-siteX(1))^2;
            a31 = (siteX(4)-siteX(1))^2;

            % Find y-variable coefficients
            a12 = (siteY(2)-siteY(1))^2;
            a22 = (siteY(3)-siteY(1))^2;
            a32 = (siteY(4)-siteY(1))^2;

            % Find z-variable coefficients
            a13 = (siteZ(2)-siteZ(1))^2;
            a23 = (siteZ(3)-siteZ(1))^2;
            a33 = (siteZ(4)-siteZ(1))^2;

            % Find values for B-matrix
            b1 = siteRad(1)^2-siteRad(2)^2-siteX(1)^2+siteX(2)^2-...
                siteY(1)^2+siteY(2)^2-siteZ(1)^2+siteZ(2)^2;
            b2 = siteRad(1)^2-siteRad(3)^2-siteX(1)^2+siteX(3)^2-...
                siteY(1)^2+siteY(3)^2-siteZ(1)^2+siteZ(3)^2;
            b3 = siteRad(1)^2-siteRad(4)^2-siteX(1)^2+siteX(4)^2-...
```

```
        siteY(1)^2+siteY(4)^2-siteZ(1)^2+siteZ(4)^2;

    % Coefficient Matrix A
    A = [a11 a12 a13; a21 a22 a23; a31 a32 a33];

    % Matrix B
    B = [b1; b2; b3];


    %Find solutions to x, y, and z
    if (D<=3)
        X = A\B;
    else
        X = pinv(A)*B;
    end

    interX(h)=X(1);
    interY(h)=X(2);
    interZ(h)=X(3);

    approxX=mean(interX);
    approxY=mean(interY);
    approxZ=mean(interZ);

    decp(h)=sqrt(approxX^2+approxY^2+approxZ^2);
end

% Calculate standard deviation of coordinate values
stdX = std(interX);
stdY = std(interY);
stdZ = std(interZ);

if (D==1)
    % Mean estimate position error per simulation
    averagePositionError1(N)=mean(decp);
    % Standard Deviation of Error
    stDevPositionError1(N)=std(decp);
    % Mean Radial Spherical Error per simulation
    MRSE1(N)= sqrt(stdX^2+stdY^2+stdZ^2);
elseif (D==2)
    % Mean estimate position error per simulation
    averagePositionError2(N)=mean(decp);
    % Standard Deviation of Error
    stDevPositionError2(N)=std(decp);
    % Mean Radial Spherical Error per simulation
    MRSE2(N)= sqrt(stdX^2+stdY^2+stdZ^2);
elseif (D==3)
    % Mean estimate position error per simulation
    averagePositionError3(N)=mean(decp);
    % Standard Deviation of Error
    stDevPositionError3(N)=std(decp);
    % Mean Radial Spherical Error per simulation
    MRSE3(N)= sqrt(stdX^2+stdY^2+stdZ^2);
elseif (D==4)
```

```
    % Mean estimate position error per simulation
    averagePositionError4(N)=mean(decp);
    % Standard Deviation of Error
    stDevPositionError4(N)=std(decp);
    % Mean Radial Spherical Error per simulation
    MRSE4(N)= sqrt(stdX^2+stdY^2+stdZ^2);
end
numSim(N) = N;

% Plot
figure(count);
clf;
hold on;
col = [1 0 0, 0 0 1, 0 1 0, 1 1 0]; %Color map
v = 1;                              %Color counter
title(['4 eNodeB, ',num2str(decp(h)),' Meters from '...
    'Approximation to UE'],'FontSize',12);
xlabel('X-Axis (meters)','FontSize',12);
ylabel('Y-Axis (meters)','FontSize',12);
zlabel('Height (meters)','FontSize',12);

for p=1:max_eNodeB
    % Create spheres
    mesh(siteX(p)+siteRad(p)*sin(th)*cos(phi),siteY(p)+...
        siteRad(p)*sin(th)*sin(phi),siteZ(p)+...
        siteRad(p)*cos(th)*ones(size(phi,1),size(phi,2)),...
        'FaceAlpha',0.2,'FaceColor',col(v:v+2),...
        'EdgeColor',col(v:v+2),'EdgeAlpha',0.1);

    % Create antenna center points
    plot3(siteX(p),siteY(p),siteZ(p),'d',...
        'MarkerFaceColor',col(v:v+2),'MarkerSize',10);

    % Create towers
    ts=round(siteZ(p));
    tx=zeros(1,ts);
    ty=zeros(1,ts);
    tz=zeros(1,ts);
    for t=0:ts
        tx(t+1)=siteX(p);
        ty(t+1)=siteY(p);
        tz(t+1)=t;
    end

    % Plot Towers
    plot3(tx,ty,tz,'-k','linewidth',4);

    v = v+3;
end
% Plot Approximation Point
  plot3(approxX,approxY,approxZ,'pr','MarkerSize',12);
  % plot(approxX,approxY,'pr','MarkerSize',12);

% Plot UE
```

```
        plot3(0,0,0,'sk','MarkerSize',12);
        hold off;
        count = count+1;
    end
end

errorMat = [averagePositionError1;averagePositionError2;...
    averagePositionError3;averagePositionError4];
MRSEMat = [MRSE1;MRSE2;MRSE3;MRSE4];

% Plot Overall Results Error of Approximation Distance to
% Actual Position
figure(count);
clf;
hold on;
title('Estimate Distance Error with 4 eNodeB','FontSize',12);
xlabel('Simulation (each with 100000 iterations)','FontSize',12);
ylabel({'Average Distance from';...
    'Approximated Location to Actual UE (meters)'},'FontSize',12);
bar(errorMat','grouped');
legend('Random Angles, Distances, Heights',...
    'Even Angles, Random Distance and Height',...
    'Even Angles, Fixed Distance, Random Height',...
    'Even Angles, Fixed Distance, Fixed Height');
hold off;

% Plot MRSE of position error
figure(count+1);
clf;
hold on;
title('MRSE with 4 eNodeB','FontSize',12);
xlabel('Simulation (each with 100000 iterations)','FontSize',12);
ylabel('MRSE 61% Containment Radius(meters)','FontSize',12);
bar(MRSEMat','grouped');
legend('Random Angles, Distances, Heights',...
    'Even Angles, Random Distance and Height',...
    'Even Angles, Fixed Distance, Random Height',...
    'Even Angles, Fixed Distance, Fixed Height');
hold off;
```

2. Random Angles, Distances and Heights Example Plots

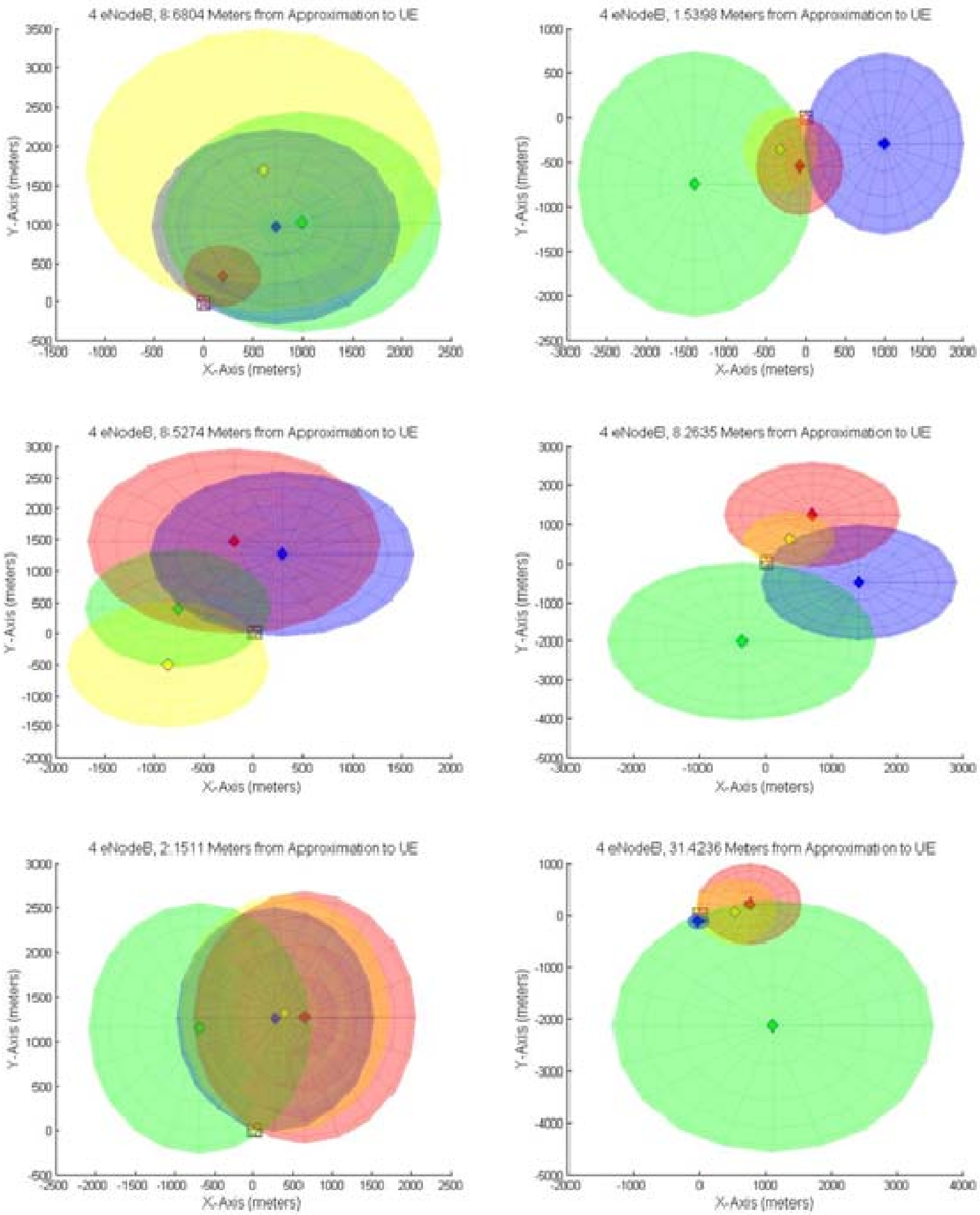


Figure 47. Sample 4 eNodeB plots with random angles, distances and heights.

(Diamonds denote the eNodeBs, a square indicates the actual UE location, and the star shows approximation center point.)

3. Even Angles with Random Distances and Heights Example Plots

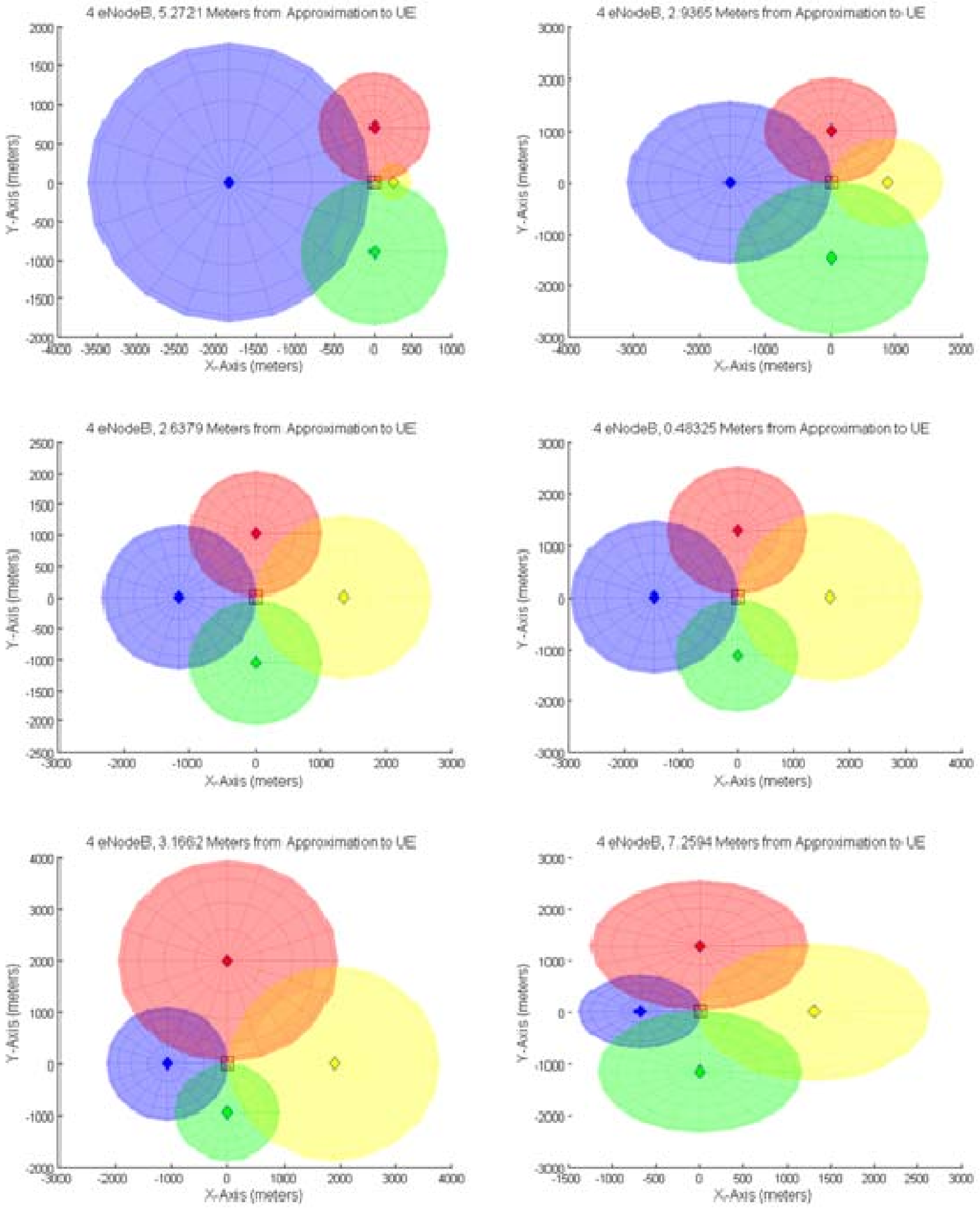
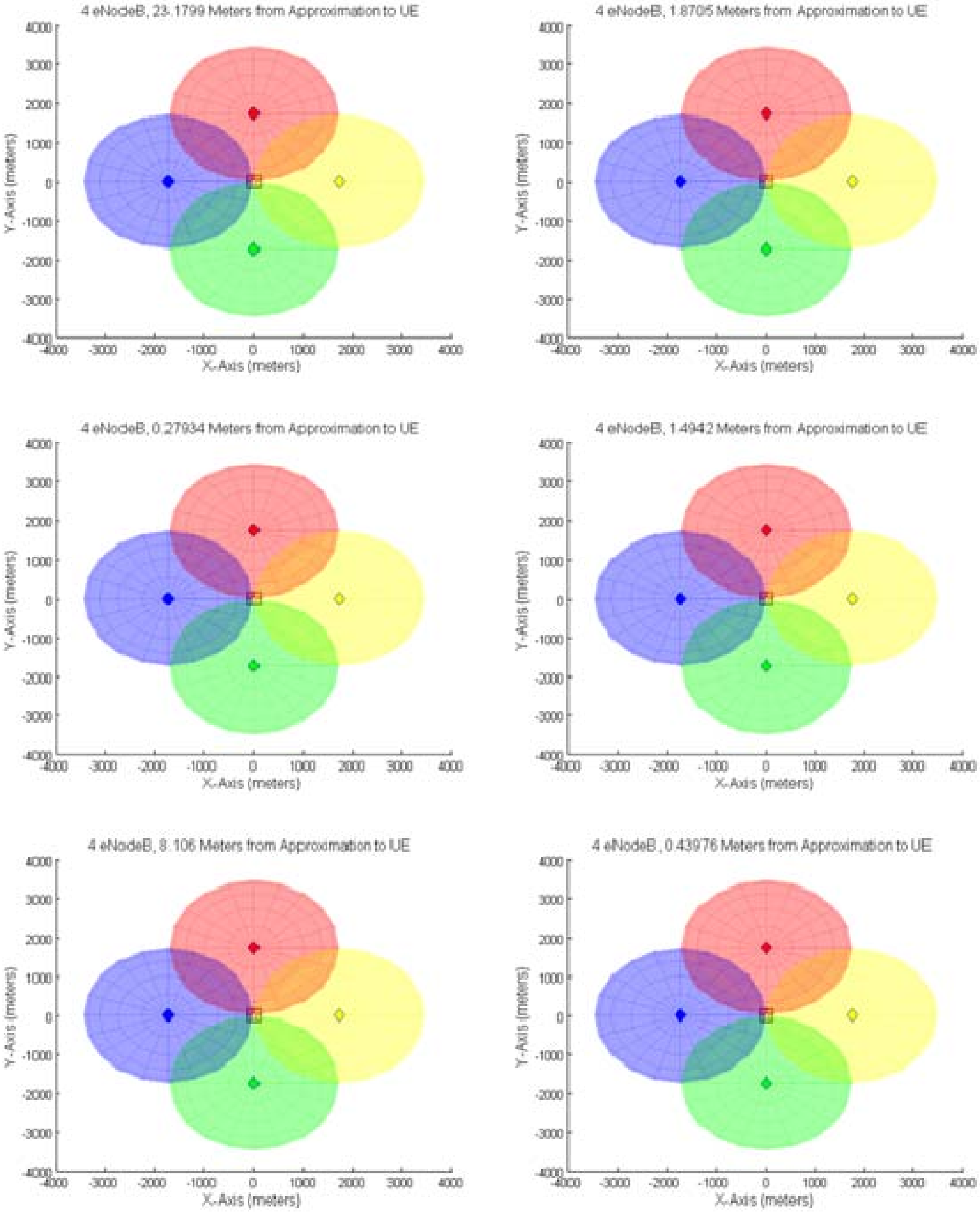


Figure 48. Sample 4 eNodeB plots with evenly spaced angles, random distances and heights.

(Diamonds denote the eNodeBs, a square indicates the actual UE location, and the star shows approximation center point.)

4. Even Angles with Fixed Distances and Random Heights Example Plots


4 eNodeB, 23.1799 Meters from Approximation to UE

4 eNodeB, 1.8705 Meters from Approximation to UE

4 eNodeB, 0.27934 Meters from Approximation to UE

4 eNodeB, 1.4942 Meters from Approximation to UE

4 eNodeB, 8.106 Meters from Approximation to UE

4 eNodeB, 0.43976 Meters from Approximation to UE

Figure 49. Sample 4 eNodeB plots with evenly spaced angles, fixed 1732m distances and random heights.

5. Even Angles with Fixed Distances and Fixed Heights Example Plots, Standard Deviation Applied

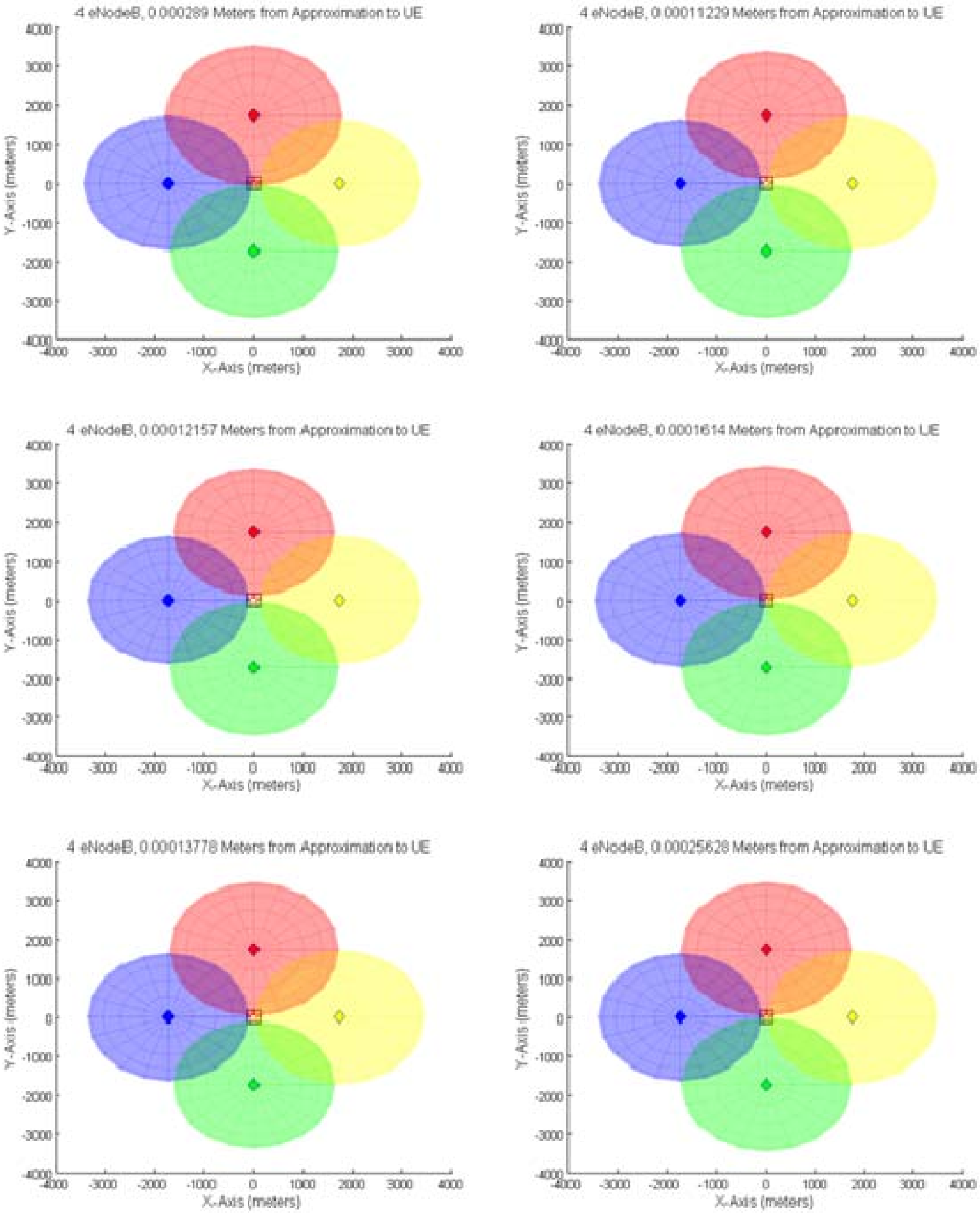


Figure 50. Sample 4 eNodeB plots with evenly spaced angles, fixed 1732m distances and fixed 305m eNodeB heights, standard deviation applied.

C. MULTIPLE BASE STATIONS

1. Multiple Base Stations MATLAB Code

```
% 3D Multiple Base Station Simulation
% Combined Simulation
% Random Angles, Distances and Heights (Fig 1-8)
% Even Angles, Random Distances and Heights (Fig 9-16)
% Even Angles, Fixed Distance of 1732m, Random Heights (Fig 17-24)
% Even Angles, Fixed Distances and Heights (Fig 25-32)
% Average Error Estimation Analysis (Fig 33)
% MRSE Error Estimation Analysis (Fig 34)
% LT L. A. Jarvis
% Rev. 6, 29 November 2010

clear all;

% LTE Calculated Timing Advance Distance Parameter
TA = 78.125;    %Meters per unit of Timing Advance
% Uncomment to apply Timing Advance standard deviation
%stdTA = 0.5;
% Uncomment to apply quantization error
%qError = 78.125;
% Uncomment to apply bias from quantization error
%bias = qError/2;

% Simulated eNodeB Distance Parameters
siteDist = 1200;    %Mean Site Distance
siteStDev = 400;    %Site Standard Deviation
siteHigh = 305;     %Mean Site Height
siteHSD = 250;      %Site Height Standard Deviation

% Simulation Parameters
max_eNodeB = 10;
iterate = 100000;

% Sphere Plots
numPoints = 10;
phi = 0:pi/numPoints:2*pi;
th =(0:pi/numPoints:pi)';

% Initialize arrays for data analysis
% Average error per simulation
averageError1 = zeros(1,max_eNodeB-3);
averageError2 = zeros(1,max_eNodeB-3);
averageError3 = zeros(1,max_eNodeB-3);
averageError4 = zeros(1,max_eNodeB-3);
% Standard dev per simulation
stDevError1 = zeros(1,max_eNodeB-3);
stDevError2 = zeros(1,max_eNodeB-3);
stDevError3 = zeros(1,max_eNodeB-3);
stDevError4 = zeros(1,max_eNodeB-3);
% Mean Radial Spherical Error
```

```
MRSE1 = zeros(1,max_eNodeB-3);
MRSE2 = zeros(1,max_eNodeB-3);
MRSE3 = zeros(1,max_eNodeB-3);
MRSE4 = zeros(1,max_eNodeB-3);
% Simulation Counter
num_eNodeB = zeros(1,max_eNodeB-3);

% Initialize counter for figures
count = 1;

for D = 1:4
    % Setup Simulation
    for N = 3:max_eNodeB
        % Initialize sized arrays
        distance = zeros(1,N); %Array of distances to eNodeBs
        siteX = zeros(1,N);    %Array of eNodeB X coordinates
        siteY = zeros(1,N);    %Array of eNodeB Y coordinates
        siteZ = zeros(1,N);    %Array of eNodeB Z coordinates
        siteTA = zeros(1,N);   %Array of Timing Advance
        siteRad = zeros(1,N);  %Array of eNodeB radii
        aX = zeros(1,N-1);       %Array of X coefficients
        aY = zeros(1,N-1);       %Array of Y coefficients
        aZ = zeros(1,N-1);       %Array of Z coefficients
        b = zeros(1,N-1);        %Array of B constants
        interX = zeros(1,iterate);
        interY = zeros(1,iterate);
        interZ = zeros(1,iterate);
      % Array of distances for center point estimation on (x,y,z) plane
        decp = zeros(1,iterate);
        % Begin iterations
        for h=1:iterate
            % Create eNodeBs
            for i=1:N
              % Distances to simulated eNodeBs with (X,Y,Z) coordinates
                if (D==1)
                    distance(i) = siteStDev*randn(1) + siteDist;
                    theta = 2*pi*rand(1);       %Random angles from UE
                    % Random eNodeB Height...Z coordinate
                    siteZ(i) = abs(siteHSD*randn(1) + siteHigh);
                elseif (D==2)
                    distance(i) = siteStDev*randn(1) + siteDist;
                    theta = 2*pi*i/N;    %Even angles from UE
                    % Random eNodeB Height...Z coordinate
                    siteZ(i) = abs(siteHSD*randn(1) + siteHigh);
                elseif (D==3)
                    distance(i) = 1732;         %Fixed distances
                    theta = 2*pi*i/N;    %Even angles from UE
                    % Random eNodeB Height...Z coordinate
                    siteZ(i) = abs(siteHSD*randn(1) + siteHigh);
                elseif (D==4)
                    distance(i) = 1732;         %Fixed distances
                    theta = 2*pi*i/N;    %Even angles from UE
                    % Fixed eNodeB Height...Z coordinate
                    siteZ(i) = siteHigh;
                end
                siteX(i) = cos(theta)*distance(i);  %X coordinate
                siteY(i) = sin(theta)*distance(i);  %Y coordinate
                siteTA(i) = round(distance(i)/TA);  %Timing Advance
                % Uncomment to apply Timing Advance standard deviation
                %siteTA(i) = round(distance(i)/TA)+...
                    %round(stdTA*randn(1));
                siteRad(i) = siteTA(i) * TA;         %Radius from TA
                % Uncomment to apply quantization error
                %siteRad(i)=siteTA(i)*TA+qError*rand(1)-bias;
            end

            % Find point of intersection

            % Due to even angles, fixed distances and fixed heights,
            % this simulation has the potential to find a case where
            % there is no solution or many solutions.  For systems
            % of equations, the unknown variables can be solved using
            % matrices in the form of AX=B, where A contains the
            % coefficients of the unknown variables and B contains
            % the constants from the individual equations.  Since the
            % coefficients for x, y and z are the same for all
            % iterations, the only values that need to be determined
            % are the constants for matrix B.  Matrix A is
            % singular, meaning the determinant of A is equal to 0.
            % Therefore, AX=B either does not exist, or is not unique.
            % For these cases, a least-squares solution is found
            % using MATLAB function "pinv" so that the simulation can
            % continue to run without interruption, and provide
            % reasonable results.

            % Find A matrix coefficient values
            for f = 2:N
                aX(f-1) = (siteX(f)-siteX(1))^2;
                aY(f-1) = (siteY(f)-siteY(1))^2;
                aZ(f-1) = (siteZ(f)-siteZ(1))^2;

                % Find B matrix constant values
                b(f-1) = siteRad(1)^2-siteRad(f)^2-siteX(1)^2+...
                    siteX(f)^2-siteY(1)^2+siteY(f)^2-siteZ(1)^2+...
                    siteZ(f)^2;
            end

            % Coefficient Matrix A
            A1 = [aX; aY; aZ;];
            A = A1';

            % Matrix B
            B = b';

            % Find solutions to x, y, and z
            if (D<=3)
                X = A\B;
            else
                X = pinv(A)*B;
                end

                interX(h)=X(1);
                interY(h)=X(2);
                interZ(h)=X(3);

                % Find the approximation coordinates
                approxX=mean(interX);
                approxY=mean(interY);
                approxZ=mean(interZ);

                % Find the positon error from approximation to UE
                if N==4
                    decp(h)=sqrt(approxX^2+approxY^2);
                else
                    decp(h)=sqrt(approxX^2+approxY^2+approxZ^2);
                end
            end

            if (D==1)
                % Mean estimate position error per simulation
                averageError1(N)=mean(decp);
                % Standard Deviation of Error
                stDevError1(N)=std(decp);
                % MRSE per simulation
                stdX=std(interX);
                stdY=std(interY);
                %stdZ=std(interZ);
                if N==4
                    stdZ=0;
                else
                    stdZ=std(interZ);
                end
                MRSE1(N)=sqrt(stdX^2+stdY^2+stdZ^2);
            elseif (D==2)
                % Mean estimate position error per simulation
                averageError2(N)=mean(decp);
                % Standard Deviation of Error
                stDevError2(N)=std(decp);
                % MRSE per simulation
                stdX=std(interX);
                stdY=std(interY);
                %stdZ=std(interZ);
                if N==4
                    stdZ=0;
                else
                    stdZ=std(interZ);
                end
                MRSE2(N)=sqrt(stdX^2+stdY^2+stdZ^2);
            elseif (D==3)
                % Mean estimate position error per simulation
                averageError3(N)=mean(decp);
                % Standard Deviation of Error
                stDevError3(N)=std(decp);
                % MRSE per simulation
                stdX=std(interX);
                stdY=std(interY);
                %stdZ=std(interZ);
                if N==4
                    stdZ=0;
                else
                    stdZ=std(interZ);
                end
                MRSE3(N)=sqrt(stdX^2+stdY^2+stdZ^2);
            elseif (D==4)
                % Mean estimate position error per simulation
                averageError4(N)=mean(decp);
                % Standard Deviation of Error
                stDevError4(N)=std(decp);
                % MRSE per simulation
                stdX=std(interX);
                stdY=std(interY);
                %stdZ=std(interZ);
                if N==4
                    stdZ=0;
                else
                    stdZ=std(interZ);
                end
                MRSE4(N)=sqrt(stdX^2+stdY^2+stdZ^2);
            end
            num_eNodeB(N) = N;

            % Plot Network
            figure(count);
            clf;
            hold on;
            grid on;
            col = [1 0 0, 0 0 1, 0 1 0, 1 1 0, 1 0 0, 0 0 1, 0 1 0,...
                1 1 0,1 0 0, 0 0 1]; %Color map
            v = 1;                   %Color counter
            zlim([0 2500]);
            view([-50 30]);
            if N==4
                title([int2str(N),' eNodeB, ',num2str(decp(h)),...
                ' Meters from Approximation to UE: Z-Coordinate ',...
                int2str(approxZ),' Meters'],'FontSize',12);
            else
                title([int2str(N),' eNodeB, ',num2str(decp(h)),...
                ' Meters from Approximation to UE'],'FontSize',12);
            end
            xlabel('X-Axis (meters)','FontSize',12);
            ylabel('Y-Axis (meters)','FontSize',12);
            zlabel('Height (meters)','FontSize',12);

            for p=1:N
                % Create spheres
                mesh(siteX(p)+siteRad(p)*sin(th)*cos(phi),siteY(p)+...
                    siteRad(p)*sin(th)*sin(phi),siteZ(p)+...
                    siteRad(p)*cos(th)*ones(size(phi,1),size(phi,2)),...
                    'FaceAlpha',0.08,'FaceColor',col(v:v+2),...
```

```
                'EdgeColor',col(v:v+2),'EdgeAlpha',0.1);

            % Create antenna center points
            plot3(siteX(p),siteY(p),siteZ(p),'d',...
                'MarkerFaceColor',col(v:v+2),'MarkerSize',10);

            % Create towers
            ts=round(siteZ(p));
            tx=zeros(1,ts);
            ty=zeros(1,ts);
            tz=zeros(1,ts);
            for t=0:ts
                tx(t+1)=siteX(p);
                ty(t+1)=siteY(p);
                tz(t+1)=t;
            end

            % Plot Towers
            plot3(tx,ty,tz,'-k','linewidth',4);

            v = v+3;
        end
        % Plot Approximation Point
        if N==4
            plot(approxX,approxY,'pr','MarkerSize',12,...
             'MarkerFaceColor','r');
        else
            plot3(approxX,approxY,approxZ,'pr','MarkerSize',12,...
             'MarkerFaceColor','r');
        end

        % Plot UE
        plot3(0,0,0,'sk','MarkerSize',12);
        hold off;

        count = count+1;
    end
end

% Plot Overall Results Error of Approximation Distance to
% Actual Position
figure(count);
clf;
hold on;
title('Estimate Error with Multiple eNodeB','FontSize',12);
xlabel('Number of eNodeB','FontSize',12);
ylabel({'Average Distance from';...
    'Center Point Estimate to UE (meters)'},'FontSize',12);
plot(num_eNodeB(3:N),averageError1(3:N),'-bs','LineWidth',2);
plot(num_eNodeB(3:N),averageError2(3:N),'-rd','LineWidth',2);
plot(num_eNodeB(3:N),averageError3(3:N),'-go','LineWidth',2);
plot(num_eNodeB(3:N),averageError4(3:N),'-mx','LineWidth',2);
legend('Random Angles, Distances, Heights',...
    'Even Angles, Random Distance and Height',...
    'Even Angles, Fixed Distance, Random Height',...
    'Even Angles, Fixed Distance, Fixed Height');
hold off;

% Plot MRSE of position error
figure(count+1);
clf;
hold on;
title('MRSE with Multiple eNodeB','FontSize',12);
xlabel('Number of eNodeB','FontSize',12);
ylabel('MRSE 61% Containment Radius (Meters)','FontSize',12);
plot(num_eNodeB(3:N),MRSE1(3:N),'-bs','LineWidth',2);
plot(num_eNodeB(3:N),MRSE2(3:N),'-rd','LineWidth',2);
plot(num_eNodeB(3:N),MRSE3(3:N),'-go','LineWidth',2);
plot(num_eNodeB(3:N),MRSE4(3:N),'-mx','LineWidth',2);
legend('Random Angles, Distances, Heights',...
    'Even Angles, Random Distance and Height',...
    'Even Angles, Fixed Distance, Random Height',...
    'Even Angles, Fixed Distance, Fixed Height');
hold off;
```

2. Random Angles, Distances and Heights Example Plots

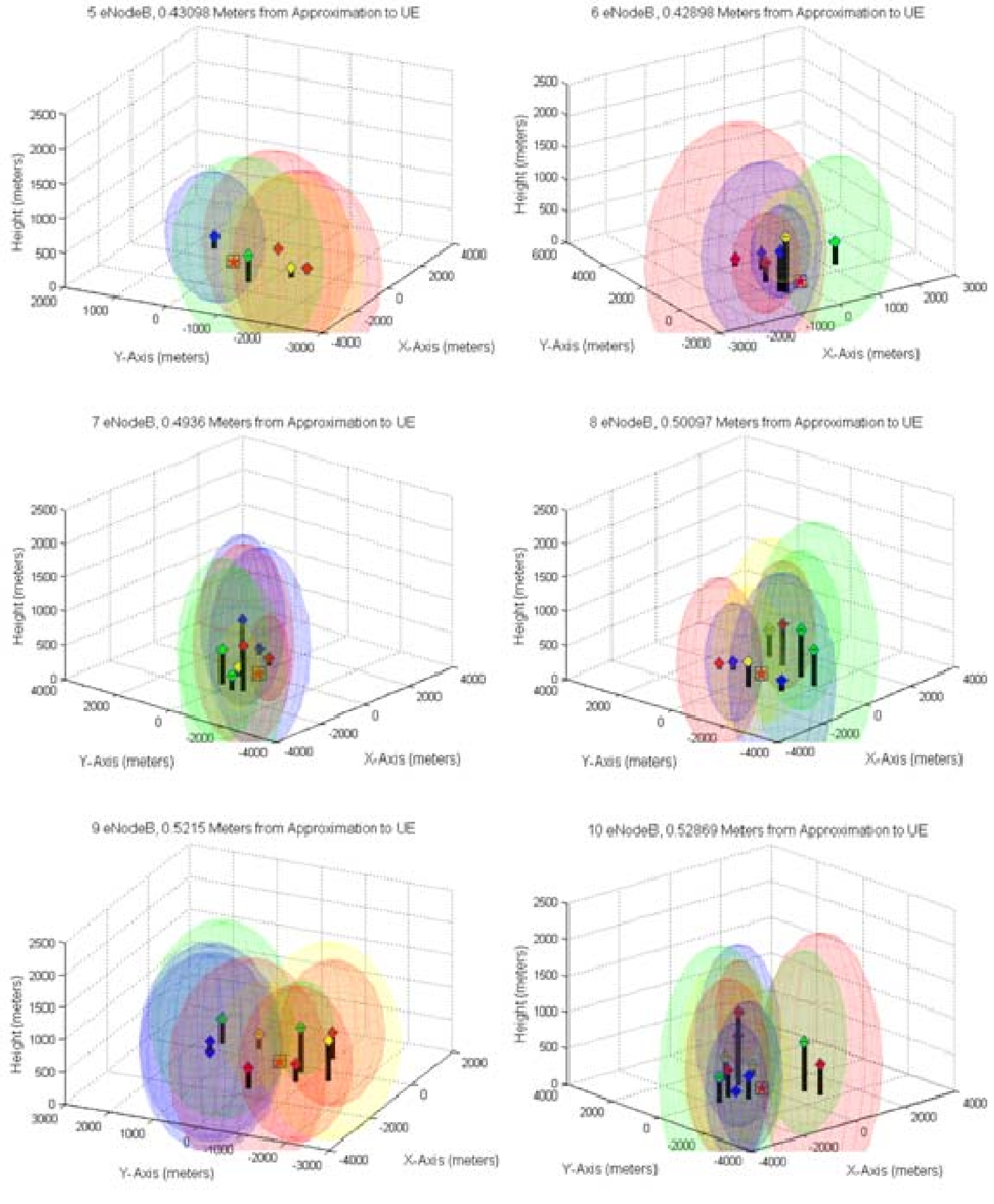
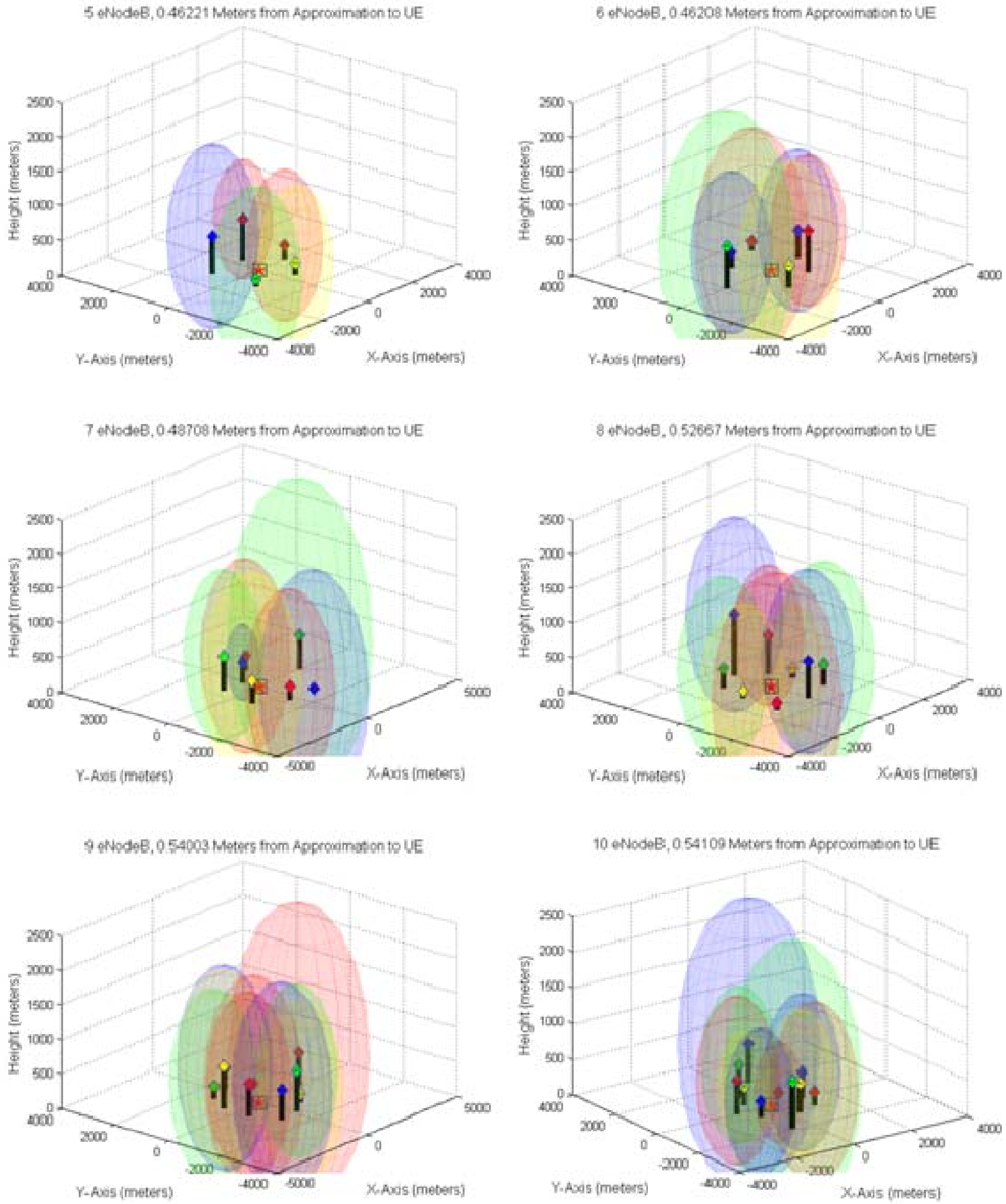


Figure 51. Sample multiple eNodeB plots with random angles, distances and heights.

(Diamonds denote the eNodeBs, a square indicates the actual UE location, and the star shows approximation center point.)

3. Even Angles with Random Distances and Heights Example Plots


5 eNodeB, 0.46221 Meters from Approximation to UE

6 eNodeB, 0.46208 Meters from Approximation to UE

7 eNodeB, 0.48708 Meters from Approximation to UE

8 eNodeB, 0.52667 Meters from Approximation to UE

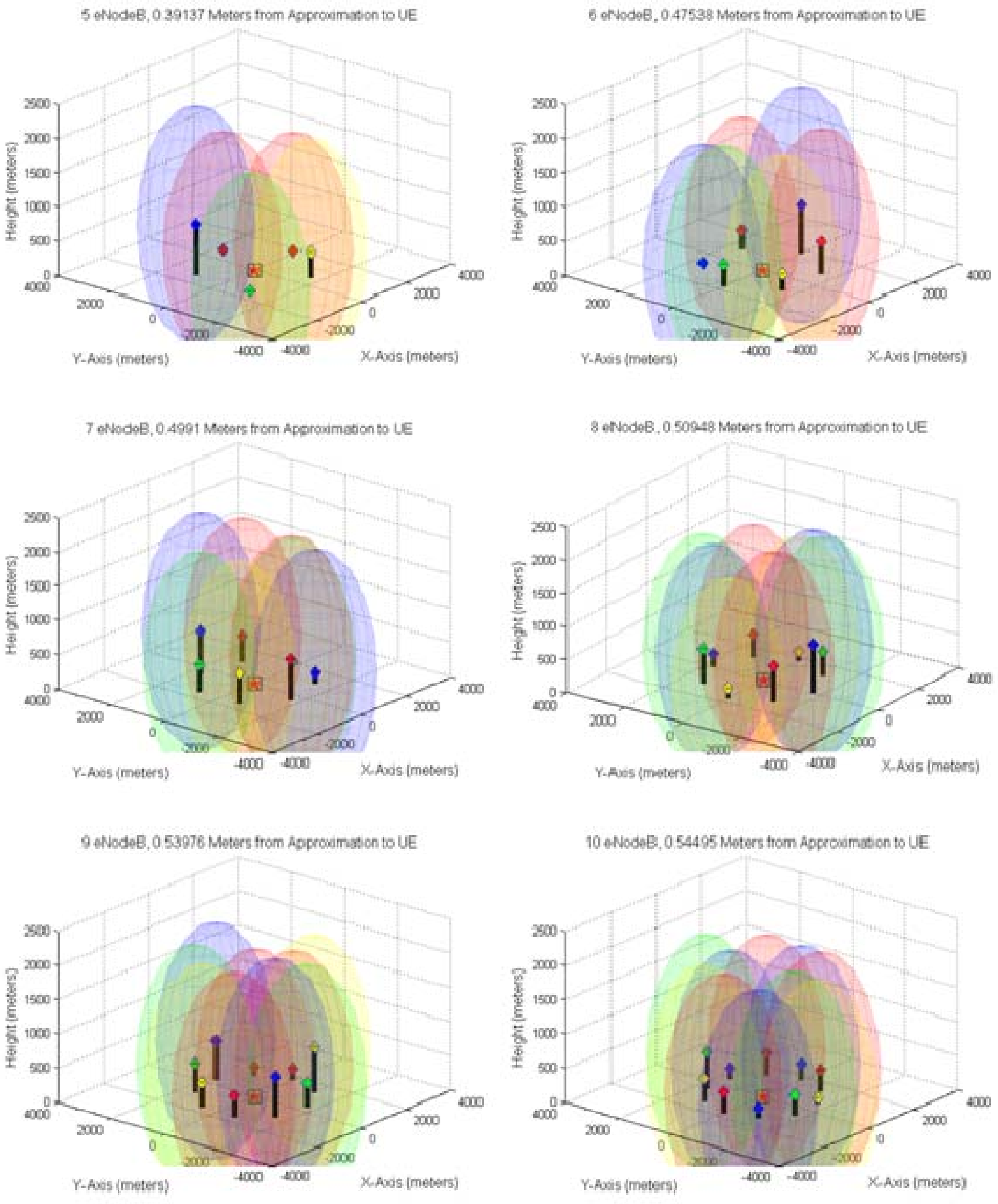
9 eNodeB, 0.54003 Meters from Approximation to UE

10 eNodeB, 0.54109 Meters from Approximation to UE

Figure 52. Sample multiple eNodeB plots with evenly spaced angles, random distances and heights.

(Diamonds denote the eNodeBs, a square indicates the actual UE location, and the star shows approximation center point.)

4. Even Angles with Fixed Distances and Random Heights Example Plots


5 eNodeB, 0.39137 Meters from Approximation to UE

6 eNodeB, 0.47538 Meters from Approximation to UE

7 eNodeB, 0.4991 Meters from Approximation to UE

8 eNodeB, 0.50948 Meters from Approximation to UE

9 eNodeB, 0.53976 Meters from Approximation to UE

10 eNodeB, 0.54495 Meters from Approximation to UE

Figure 53. Sample multiple eNodeB plots with evenly spaced angles, fixed 1732m distances and random heights.

5. Even Angles with Fixed Distances and Fixed Heights Example Plots

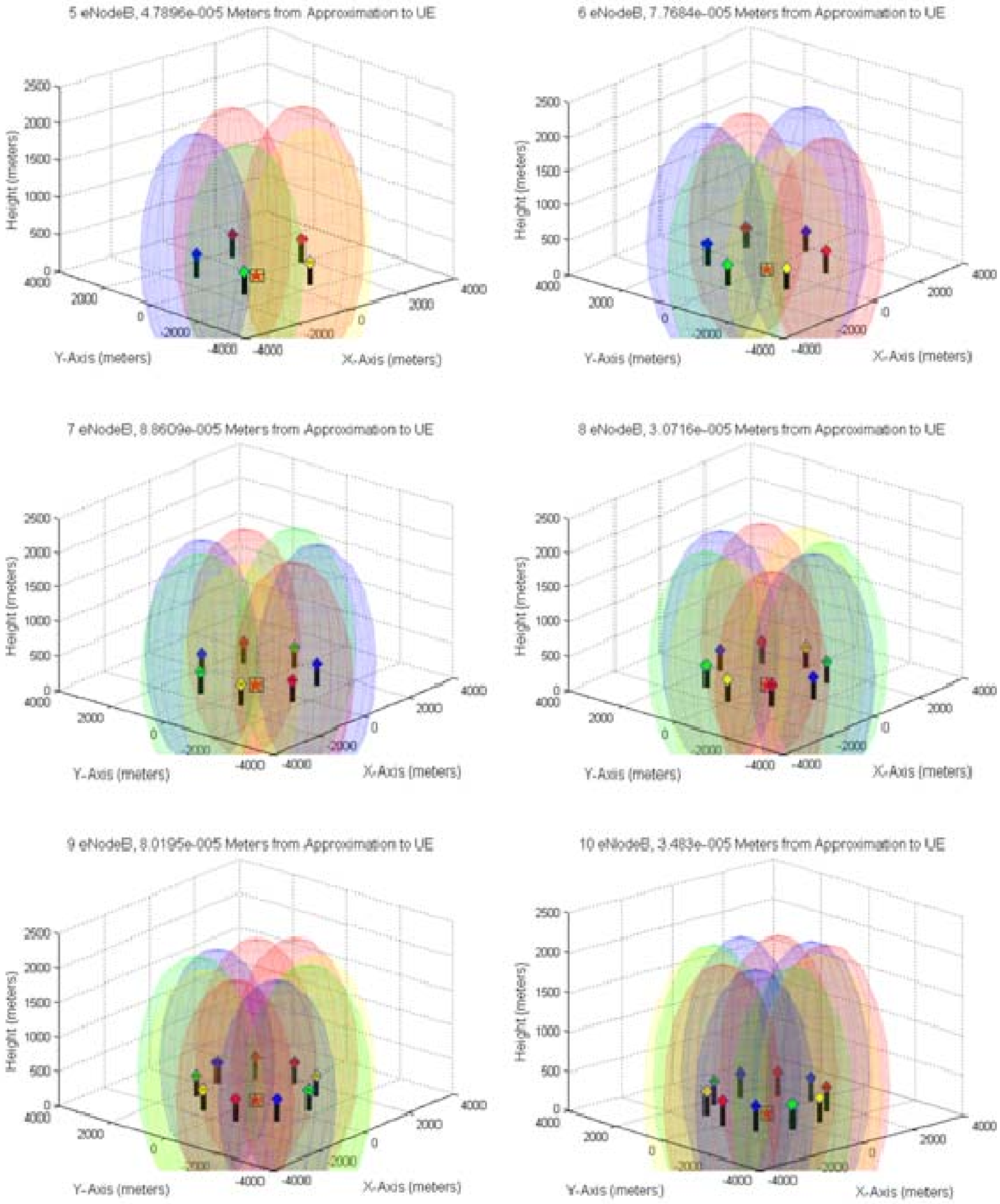


Figure 54. Sample multiple eNodeB plots with evenly spaced angles, fixed 1732m distances and fixed 305m eNodeB heights.

LIST OF REFERENCES

[1] E. Dahlman, S. Parkvall, J. Skold and P. Beming, *3G Evolution: HSPA and LTE for Mobile Broadband, Second Edition*. Burlington, Massachusetts: Elsevier Ltd., pp. 3–10, 491, 2008.

[2] S. Sesia, I. Toufik, M. Baker, *LTE-The UMTS Long Term Evolution From Theory to Practice.* West Sussex, United Kingdom: John Wiley & Sons Ltd., pp. 7–10, 2009.

[3] S. Parkvall, E. Dahlman, A. Furuskar, Y. Jading, M. Olsson, S. Wanstedt, and K. Zangi, "LTE-Advanced-evolving LTE towards IMT-Advanced," in *Vehicular Technology Conference*, pp. 1–5, 2008.

[4] International Telecommunications Union. (2010, 10/21). ITU paves way for next-generation 4G mobile technologies. [Online]. 2010(10/29), Available: http://www.itu.int/net/pressoffice/press_releases/2010/40.aspx

[5] LTE World Forum. (2010, October). LTE maps. [Online]. 2010(10/13), Available: http://www.ltemaps.org/

[6] Federal Communications Commission, "FCC amended report to congress on the deployment of E-911 Phase II services Tier III Service Providers," April 2005.

[7] Federal Communications Commission, "Public Safety and Homeland Security Bureau Seeks Comment on Petitions for Waiver to Deploy 700 MHz Public Safety Broadband Networks," PS Docket No. 06-229, August 2009.

[8] D. Barber, "Geolocation of WiMAX subscribers stations based on the timing adjust ranging parameter," M.S. thesis, Naval Postgraduate School, Monterey, California, December 2009.

[9] H. H. Loomis, "Geolocation of electromagnetic emitters," Naval Postgraduate School, Monterey, California, November 2009.

[10] M. A. Spirito and A. G. Mattioli, "Preliminary experimental results of a GSM mobile phones positioning system based on timing advance," in *Vehicular Technology Conference*, pp. 2072–2076, 1999.

[11] G. P. Yoast and S. Panchapakesan, "Improvement in estimation of time of arrival (TOA) from timing advance (TA)," in *IEEE 1998 International Conference on Universal Personal Communications*, pp. 1367–1372, 1998.

[12] Agilent Technologies. (2009). LTE_DL_Src (Downlink Baseband Signal Source). [Online]. 2010(10/15), Available: http://edocs.soco.agilent.com/display/ads2009/LTE+DL+Src+(Downlink+Baseband+Signal+Source)

[13] C. Eklund, R. B. Marks, K. L. Stanwood and S. Wang, “IEEE Standard 802.16: A technical overview of the WirelessMAN Air Interface for Broadband Wireless Access,” in *IEEE Communications Magazine*, pp. 98–107, June 2002.

[14] J. Zyren, “Overview of the 3GPP Long Term Evolution Physical Layer,” white paper, Freescale Semiconductor, Inc., July 2007.

[15] 36.211 3GPP E-UTRA Physical Channels and Modulation, Release 9 (2010-03).

[16] 36.321 3GPP E-UTRA Medium Access Control (MAC) protocol specification, Release 9 (2010-06).

[17] 36.213 3GPP E-UTRA Physical layer procedures, Release 9 (2010-06).

[18] R. Whitty, “Three-dimensional geolocation of Mobile WiMAX subscribers,” M.S. thesis, Naval Postgraduate School, Monterey, California, December 2010.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Chair, Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

4. Professor John McEachen
   Naval Postgraduate School
   Monterey, California

5. Professor Herschel Loomis
   Naval Postgraduate School
   Monterey, California

6. LT Leslie Jarvis
   Naval Postgraduate School
   Monterey, California